

Panduan Pengguna

AWS CloudShell



AWS CloudShell: Panduan Pengguna

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan kekayaan masing-masing pemiliknya, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

Table of Contents

Apa itu AWS CloudShell?	1
AWS CloudShell fitur	1
AWS Command Line Interface	2
Kerang dan alat pengembangan	2
Penyimpanan tetap	2
Keamanan	3
Opsi kustomisasi	3
Pemulihan sesi	3
Harga untuk AWS CloudShell	4
Bagaimana cara memulainya AWS CloudShell?	4
AWS CloudShell Topik utama	7
FAQ	7
Bagaimana cara saya mulai menggunakan AWS CloudShell?	8
Apa yang perlu saya akses AWS CloudShell?	8
Apa yang AWS CloudShell ada diConsole Toolbar?	8
Bagaimana cara saya meluncurkan AWS CloudShell diConsole Toolbar?	9
Yang Wilayah AWS CloudShell tersedia di?	9
Yang Wilayah AWS ditetapkan jika AWS CloudShell tidak tersedia di Wilayah yang dipilih saat Anda meluncurkan CloudShell diConsole Toolbar?	9
Jenis cangkang apa yang bisa saya gunakan AWS CloudShell?	9
Browser web apa yang dapat saya gunakan AWS CloudShell?	9
Bagaimana cara membuat dan mengelola AWS CloudShell lingkungan saya?	9
Browser web mana yang dapat saya gunakan ketika saya meluncurkan AWS CloudShell diConsole Toolbar?	10
Dapatkah saya mengunduh file ketika saya meluncurkan AWS CloudShell diConsole Toolbar?	10
Perangkat lunak apa yang sudah diinstal sebelumnya di lingkungan shell saya?	10
Bisakah saya menginstal perangkat lunak yang tidak tersedia di lingkungan shell?	11
Bisakah saya membatasi tindakan yang dapat dilakukan pengguna? AWS CloudShell	11
Bagaimana saya bisa memindahkan data dari direktori home saya jika saya ingin mengubah Wilayah AWS tempat saya menggunakan AWS CloudShell?	11
Dapatkah saya meningkatkan batas yang menentukan kapan AWS CloudShell waktu habis karena pengguna tidak aktif?	11

Dapatkan saya mengakses AWS CloudShellAWS Console Mobile Application dari layar beranda?	12
Bagaimana saya bisa meluncurkan AWS CloudShell di AWS Console Mobile Application? ...	12
Bisakah saya menggunakan tombol pengubah di keyboard iOS dan Android saya saat menggunakan AWS CloudShell di? AWS Console Mobile Application	12
Bisakah saya membagi tampilan AWS CloudShell tab menjadi beberapa tab di? AWS Console Mobile Application	12
Dapatkan saya mengakses AWS CloudShellConsole Toolbar di perangkat seluler?	12
Mulai	13
Prasyarat	13
Daftar Isi	14
Langkah 1: Masuk ke AWS Management Console	14
Langkah 2: Pilih Wilayah, luncurkanAWS CloudShell, dan pilih shell	17
Langkah 3: Unduh file dari AWS CloudShell	20
Langkah 4: Unggah file ke AWS CloudShell	21
Langkah 5: Hapus file dari AWS CloudShell	22
Langkah 6: Buat cadangan direktori home	22
Langkah 7: Mulai ulang sesi shell	24
Langkah 8: Hapus direktori home sesi shell	25
Langkah 9: Edit kode file Anda dan jalankan menggunakan baris perintah	26
Langkah 10: Gunakan AWS CLI untuk menambahkan file sebagai objek di bucket Amazon S3	28
Topik terkait	29
Tutorial	30
Tutorial: Menyalin beberapa file	30
Mengunggah dan mengunduh beberapa file menggunakan Amazon S3	31
Mengunggah dan mengunduh beberapa file menggunakan folder zip	34
Tutorial: Menggunakan CodeCommit	35
Prasyarat	36
Langkah 1: Membuat dan mengklon CodeCommit repositori	36
Langkah 2: Tahap dan komit file sebelum mendorongnya ke CodeCommit repositori Anda ...	37
Tutorial: Membuat URL	38
Prasyarat	38
Langkah 1: Buat peran IAM untuk memberikan akses ke bucket Amazon S3	39
Buat URL	40

Tutorial: Membangun wadah Docker di dalam AWS CloudShell dan mendorong ke Amazon ECR	41
Prasyarat	42
Prosedur Tutorial	42
Bersihkan	44
Tutorial: Menyebarkan fungsi Lambda menggunakan AWS CDK	44
Prasyarat	44
Prosedur Tutorial	44
Bersihkan	47
Bekerja dengan AWS CloudShell	48
Menavigasi antarmuka AWS CloudShell	48
.....	48
Bekerja di Wilayah AWS	49
Menentukan default Wilayah AWS Anda untuk AWS CLI	50
Bekerja dengan file dan penyimpanan	51
Bekerja dengan Docker	52
Fitur aksesibilitas	53
Navigasi keyboard diCloudShell	53
CloudShellfitur aksesibilitas terminal	53
Memilih ukuran font dan tema antarmuka diCloudShell	53
Bekerja denganAWS Layanan	55
AWS CLIcontoh baris perintah untukAWS layanan yang dipilih	55
DynamoDB	56
AWS Cloud9	56
Amazon EC2	57
S3 Glacier	57
AWSElastic Beanstalk	57
Amazon ECS	58
CLI AWS SAM	58
MenyesuaikanAWS CloudShell	59
Memisahkan tampilan baris perintah menjadi beberapa tab	59
Mengubah ukuran font	60
Mengubah tema antarmuka	60
Menggunakan Safe Paste untuk teks multiline	60
Menggunakanntmuxke pemulihan sesi	61
Keamanan	3

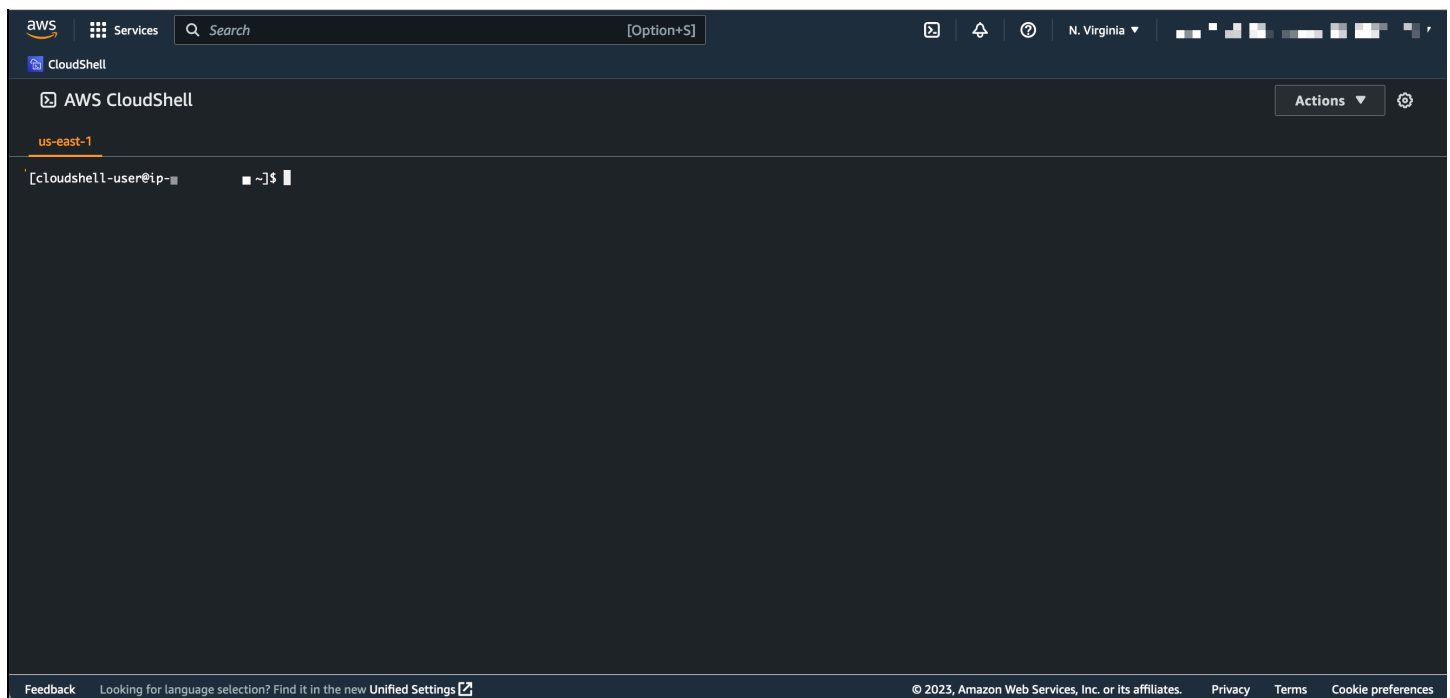
Perlindungan data	63
Enkripsi data	64
Manajemen Identitas dan Akses	64
Audiens	65
Mengautentikasi dengan identitas	66
Mengelola akses menggunakan kebijakan	69
Bagaimana AWS CloudShell bekerja dengan IAM	72
Contoh kebijakan berbasis identitas	79
Memecahkan masalah	82
Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM	85
Pencatatan dan pemantauan	91
Memantau aktivitas dengan CloudTrail	91
AWS CloudShell di CloudTrail	92
Validasi kepatuhan	94
Ketangguhan	99
Keamanan infrastruktur	100
Konfigurasi dan analisis kerentanan	100
Praktik terbaik keamanan	101
FAQ Keamanan	101
Apa saja AWS proses dan teknologi yang digunakan saat Anda meluncurkan CloudShell dan memulai sesi shell?	102
Apakah mungkin untuk membatasi akses jaringan? CloudShell	102
Dapatkah saya menyesuaikan CloudShell lingkungan saya?	102
Di mana \$HOME direktori saya sebenarnya disimpan di AWS Cloud?	102
Apakah mungkin untuk mengenkripsi \$HOME direktori saya?	103
Bisakah saya menjalankan pemindaian virus di \$HOME direktori saya?	103
AWS CloudShell lingkungan komputasi	104
Menghitung sumber daya lingkungan	104
CloudShell persyaratan jaringan	104
Perangkat lunak pra-instal	105
Kerang	106
AWS antarmuka baris perintah (CLI)	106
Runtime dan AWS SDK: Node.js dan Python 3	110
Alat pengembangan dan utilitas shell	113
Instalasi AWS CLI ke direktori home Anda	120
Menginstal perangkat lunak pihak ketiga di lingkungan shell Anda	122

Memodifikasi shell Anda dengan skrip	123
Migrasi dari Amazon Linux 2 ke Amazon Linux 2023	124
AWS CloudShellFAQ Migrasi	124
Pemecahan masalah	126
Memecahkan masalah kesalahan	126
Tidak dapat memulai lingkungan. Untuk mencoba lagi, refresh browser atau restart dengan memilih Actions, Restart AWS CloudShell	127
Tidak dapat memulai lingkungan. Anda tidak memiliki izin yang diperlukan. Minta administrator IAM Anda untuk memberikan akses ke AWS CloudShell	127
Tidak dapat mengakses AWS CloudShell baris perintah	127
Tidak dapat melakukan ping ke alamat IP eksternal	127
Ada beberapa masalah dalam mempersiapkan terminal Anda	128
Tombol panah tidak berfungsi dengan benar di PowerShell	128
Soket Web yang tidak didukung menyebabkan kegagalan untuk memulai sesi CloudShell ..	129
Tidak dapat mengimpor AWSPowerShell.NetCore modul	130
Docker tidak berjalan saat menggunakan AWS CloudShell	131
Docker kehabisan ruang disk	132
docker pushwaktunya habis dan terus mencoba lagi	132
Peramban yang didukung	133
Wilayah yang Didukung	134
GovCloud Daerah	134
Wilayah Keikutsertaan	135
Wilayah yang Didukung untuk Docker	135
Service Quotas dan batasan	136
Penyimpanan tetap	136
Penggunaan bulanan	137
Ukuran perintah	137
Kerang serentak	137
Sesi Shell	138
Akses jaringan dan transfer data	138
Pembatasan pada file sistem dan isi ulang halaman	138
Riwayat dokumen	140
.....	cxliii

Apa itu AWS CloudShell?

AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda luncurkan langsung dari file. AWS Management Console Anda dapat menavigasi ke CloudShell dari AWS Management Console beberapa cara yang berbeda. Untuk informasi selengkapnya, lihat [Cara memulai AWS CloudShell?](#)

Anda dapat menjalankan AWS CLI perintah menggunakan shell pilihan Anda, seperti Bash, PowerShell, atau Z shell. Dan Anda dapat melakukan ini tanpa mengunduh atau menginstal alat baris perintah.



Saat Anda meluncurkan AWS CloudShell, [lingkungan komputasi](#) yang didasarkan pada Amazon Linux 2023 dibuat. Dalam lingkungan ini, Anda dapat mengakses [berbagai alat pengembangan pra-instal](#), opsi untuk [mengunggah](#) dan [mengunduh](#) file, dan [penyimpanan file yang bertahan](#) di antara sesi.

(Coba sekarang: [Memulai dengan AWS CloudShell](#))

AWS CloudShell fitur

Topik ini menjelaskan cara memulai CloudShell dari Konsol, beralih secara mulus di antara shell baris perintah pilihan Anda, dan CloudShell menyesuaikan dengan preferensi Anda. Selain itu, Anda dapat

menggunakan hingga 1 GB penyimpanan persisten di masing-masing Wilayah AWS, dan bagaimana CloudShell lingkungan dilindungi oleh fitur keamanan tertentu.

AWS Command Line Interface

Anda dapat meluncurkan AWS CloudShell dari AWS Management Console. AWS Kredensi yang Anda gunakan untuk masuk ke konsol secara otomatis tersedia di sesi shell baru. Karena AWS CloudShell pengguna telah diautentikasi sebelumnya, Anda tidak perlu mengonfigurasi kredensial saat berinteraksi dengan menggunakan versi 2. Layanan AWS CLI AWS CLI Ini sudah diinstal sebelumnya di lingkungan komputasi shell.

Untuk informasi selengkapnya tentang berinteraksi dengan Layanan AWS menggunakan antarmuka baris perintah, lihat [Bekerja dengan AWS layanan di AWS CloudShell](#).

Kerang dan alat pengembangan

Dengan shell yang dibuat untuk AWS CloudShell sesi, Anda dapat beralih dengan mulus di antara shell baris perintah pilihan Anda. Lebih khusus lagi, Anda dapat beralih antara Bash, PowerShell, dan Z shell. Anda juga memiliki akses ke alat dan utilitas pra-instal. Ini termasuk git, make, pip, sudo, tar, tmux, vim, wget, dan zip.

Lingkungan shell sudah dikonfigurasi sebelumnya dengan dukungan untuk beberapa bahasa perangkat lunak utama terkemuka, seperti Node.js dan Python. Ini berarti, misalnya, Anda dapat menjalankan Node.js dan Python memproyeksikan tanpa terlebih dahulu melakukan instalasi runtime. PowerShell pengguna dapat menggunakan .NET Core runtime.

Anda dapat melakukan komit file yang dibuat atau diunggah AWS CloudShell ke repositori lokal sebelum mendorong file-file ini ke repositori jarak jauh yang dikelola oleh AWS CodeCommit

Untuk informasi selengkapnya, lihat [AWS CloudShell lingkungan komputasi: spesifikasi dan perangkat lunak](#).

Penyimpanan tetap

Dengan AWS CloudShell, Anda dapat menggunakan hingga 1 GB penyimpanan persisten Wilayah AWS di masing-masing tanpa biaya tambahan. Penyimpanan persisten terletak di direktori home Anda (\$HOME) dan bersifat pribadi untuk Anda. Tidak seperti sumber daya lingkungan sementara yang didaur ulang setelah setiap sesi shell berakhir, data di direktori home Anda tetap ada di antara sesi.

Untuk informasi selengkapnya tentang retensi data dalam penyimpanan persisten, lihat [Penyimpanan tetap](#).

Keamanan

AWS CloudShell Lingkungan dan penggunaannya dilindungi oleh fitur keamanan tertentu. Ini termasuk fitur seperti manajemen izin IAM, pembatasan sesi shell, dan Safe Paste untuk input teks.

Manajemen izin dengan IAM

Sebagai administrator, Anda dapat memberikan dan menolak izin kepada AWS CloudShell pengguna yang menggunakan kebijakan IAM. Anda juga dapat membuat kebijakan yang menentukan tindakan tertentu yang dapat dilakukan pengguna dengan lingkungan shell. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

Manajemen sesi Shell

Sesi yang tidak aktif dan berjalan lama secara otomatis dihentikan dan didaur ulang. Untuk informasi selengkapnya, lihat [Sesi Shell](#).

Tempel Aman untuk masukan teks

Safe Paste diaktifkan secara default. Fitur keamanan ini mengharuskan Anda memverifikasi bahwa teks multiline yang ingin Anda tempelkan ke shell tidak mengandung skrip berbahaya. Untuk informasi selengkapnya, lihat [Menggunakan Safe Paste untuk teks multiline](#).

Opsi kustomisasi

Anda dapat menyesuaikan AWS CloudShell pengalaman Anda dengan preferensi Anda yang tepat. Misalnya, Anda dapat mengubah tata letak layar (beberapa tab), ukuran teks yang ditampilkan, dan beralih di antara tema antarmuka terang dan gelap. Untuk informasi selengkapnya, lihat [Menyesuaikan AndaAWS CloudShellpengalaman](#).

Anda juga dapat memperluas lingkungan shell Anda dengan [menginstal perangkat lunak Anda sendiri](#) dan [memodifikasi skrip shell start-up](#).

Pemulihan sesi

Fungsi pemulihan sesi mengembalikan sesi yang Anda jalankan di satu atau beberapa tab browser di CloudShell terminal. Jika Anda menyegarkan atau membuka kembali tab browser yang baru saja ditutup, fungsi ini melanjutkan sesi hingga shell dihentikan karena sesi tidak aktif. Untuk terus

menggunakan CloudShell sesi Anda, tekan tombol apa saja di dalam jendela terminal. Untuk informasi selengkapnya tentang sesi Shell, lihat [sesi Shell](#).

Pemulihan sesi juga mengembalikan output terminal terbaru dan menjalankan proses di setiap tab terminal.

Note

Pemulihan sesi tidak tersedia di aplikasi seluler.

Harga untuk AWS CloudShell

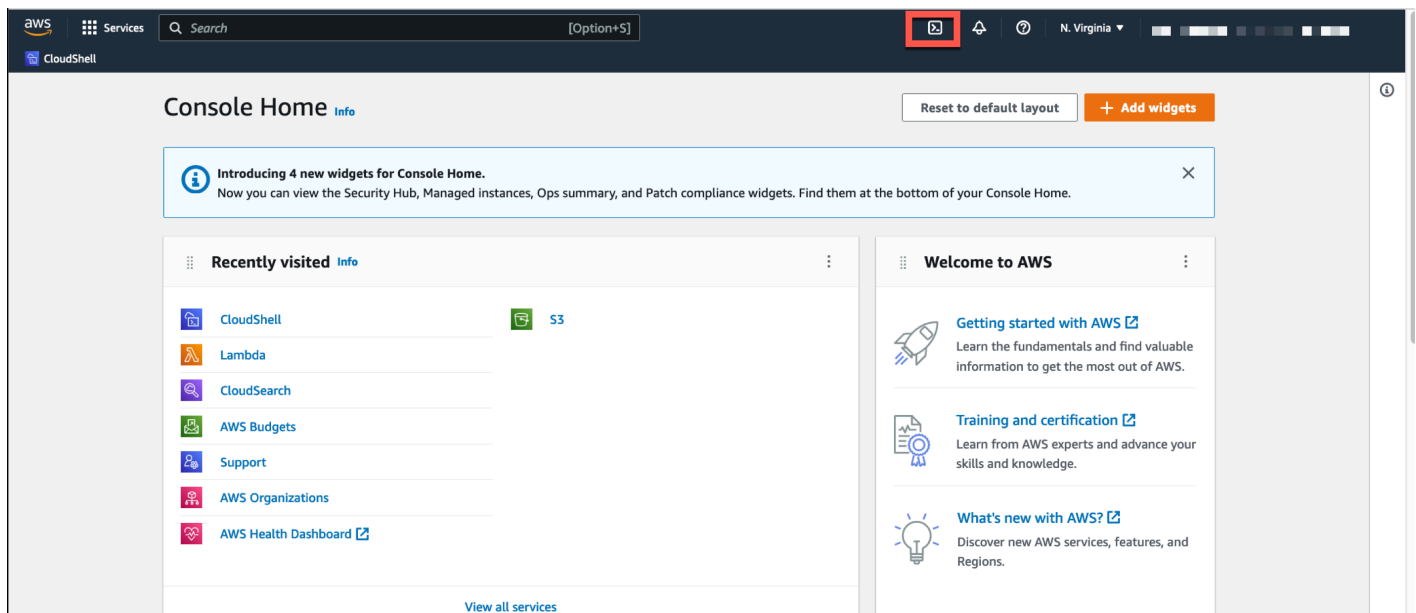
AWS CloudShell adalah Layanan AWS yang tersedia tanpa biaya tambahan. Namun, Anda membayar AWS sumber daya lain yang Anda jalankan AWS CloudShell. Selain itu, [tarif transfer data standar](#) juga berlaku. Untuk informasi selengkapnya, lihat [harga AWS CloudShell](#).

Untuk informasi selengkapnya, lihat [Kuota layanan dan pembatasan untuk AWS CloudShell](#).

Bagaimana cara memulainya AWS CloudShell?

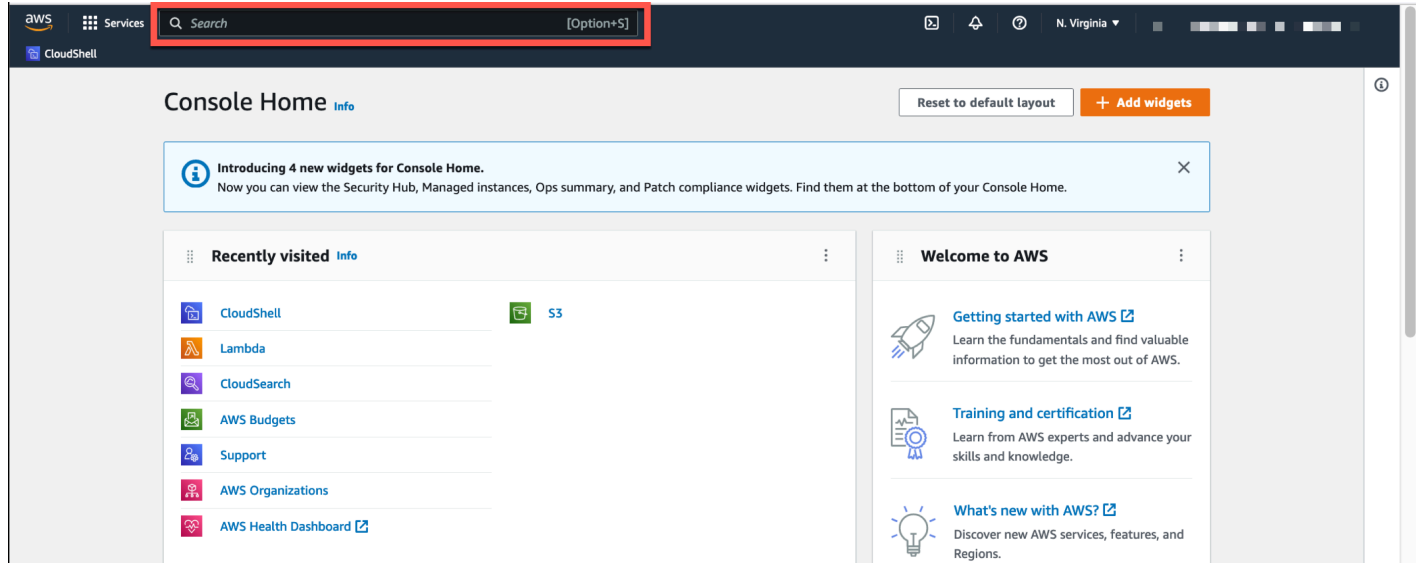
Untuk mulai bekerja dengan shell, masuk ke AWS Management Console dan pilih salah satu opsi berikut:

- Pada bilah navigasi, pilih CloudShell ikon.



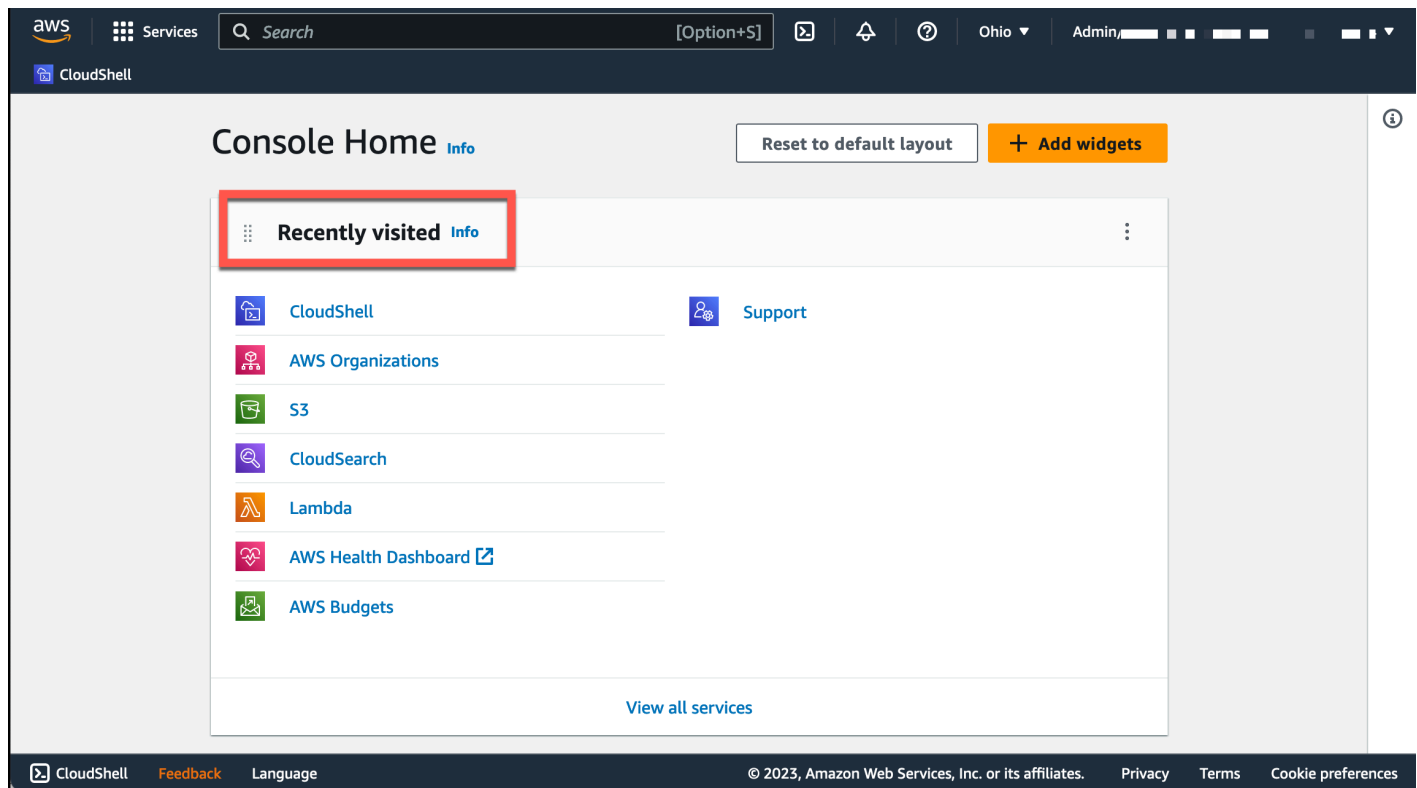
- Di kotak Pencarian, ketik “CloudShell”, lalu pilih CloudShell.

Langkah ini membuka CloudShell sesi Anda ke layar penuh.

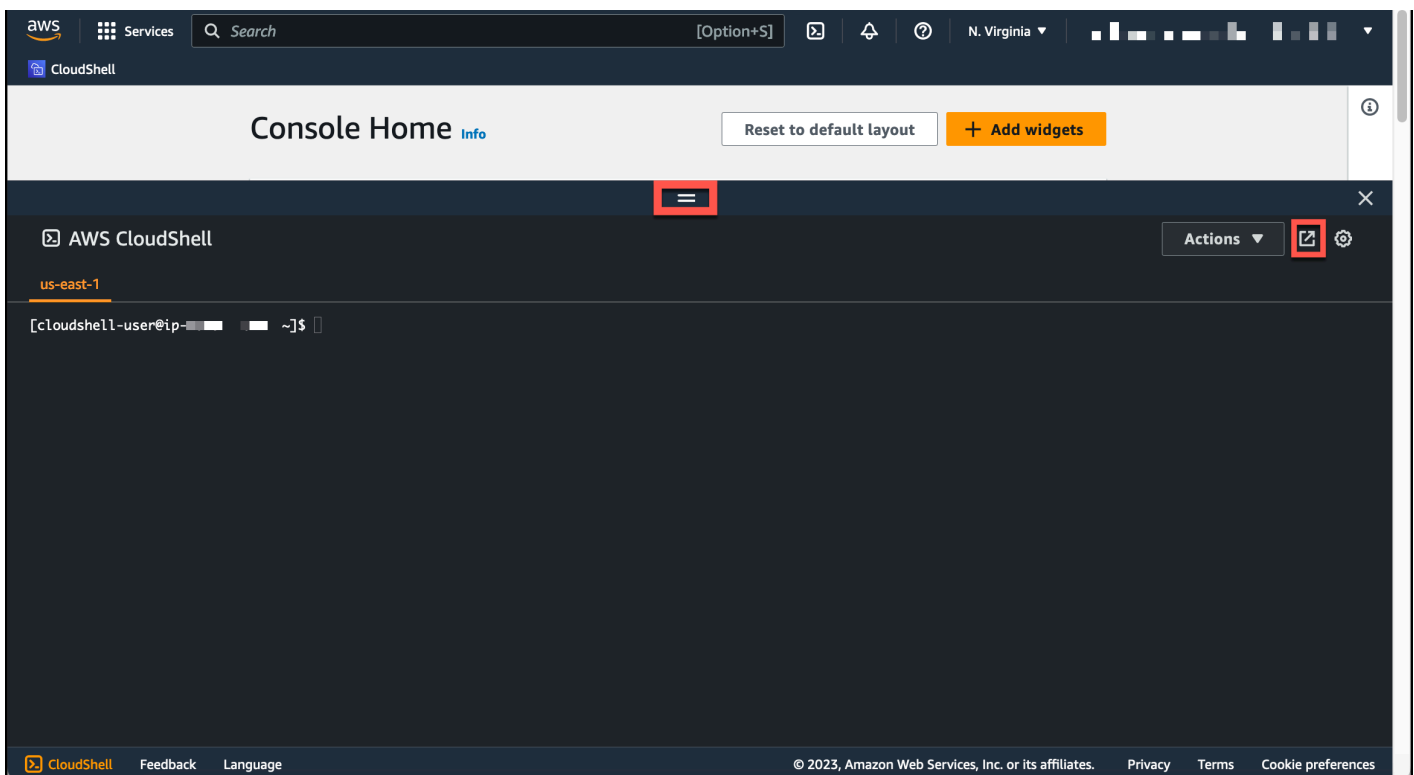
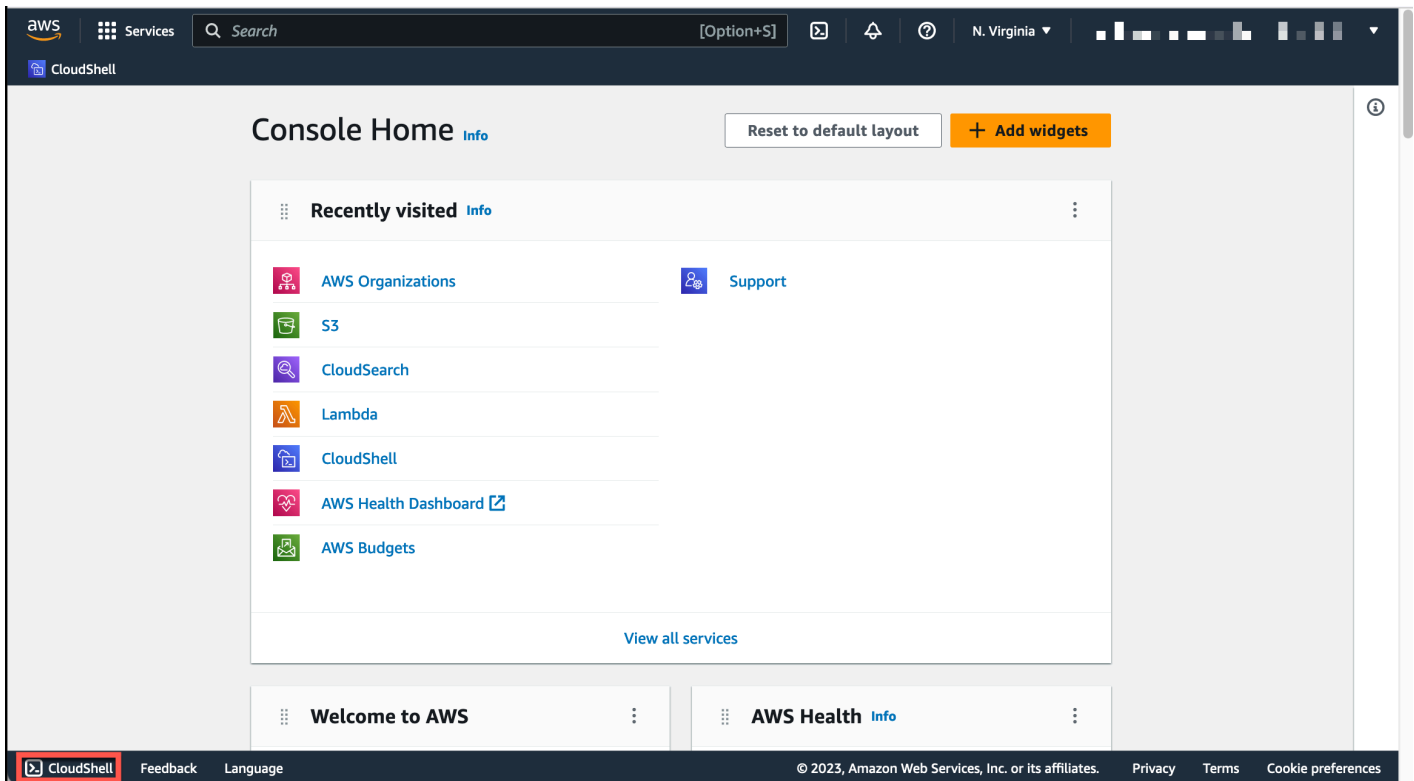


- Di widget yang baru dikunjungi, pilih CloudShell.

Langkah ini membuka CloudShell sesi Anda ke layar penuh.



- Pilih CloudShell pada Console Toolbar, di kiri bawah konsol. Anda dapat menyesuaikan ketinggian CloudShell sesi Anda dengan menyeret=.



Anda juga dapat mengalihkan CloudShell sesi Anda ke layar penuh dengan mengklik Buka di tab browser baru.

Untuk petunjuk tentang cara masuk ke AWS Management Console dan melakukan tugas-tugas utama dengan AWS CloudShell, lihat [Memulai dengan AWS CloudShell](#).

AWS CloudShell Topik utama

- [Memulai dengan AWS CloudShell](#)
- [Bekerja dengan AWS CloudShell](#)
- [Bekerja dengan AWS layanan di AWS CloudShell](#)
- [Menyesuaikan Anda AWS CloudShell pengalaman](#)
- [AWS CloudShell lingkungan komputasi: spesifikasi dan perangkat lunak](#)

AWS CloudShell FAQ

Berikut ini adalah jawaban atas beberapa pertanyaan umum tentang AWS CloudShell.

Untuk lebih banyak FAQ yang berfokus pada keamanan, lihat [AWS CloudShell FAQ Keamanan](#)

- [Bagaimana cara saya mulai menggunakan AWS CloudShell?](#)
- [Apa yang perlu saya akses AWS CloudShell?](#)
- [Apa yang AWS CloudShell ada di Console Toolbar?](#)
- [Bagaimana cara saya meluncurkan AWS CloudShell di Console Toolbar?](#)
- [Bagaimana cara membuat dan mengelola AWS CloudShell lingkungan saya?](#)
- [Yang Wilayah AWS AWS CloudShell tersedia di?](#)
- [Yang Wilayah AWS ditetapkan jika AWS CloudShell tidak tersedia di Wilayah yang dipilih saat Anda meluncurkan CloudShell di Console Toolbar?](#)
- [Jenis cangkang apa yang bisa saya gunakan AWS CloudShell?](#)
- [Browser web apa yang dapat saya gunakan AWS CloudShell?](#)
- [Browser web mana yang dapat saya gunakan ketika saya meluncurkan AWS CloudShell di Console Toolbar?](#)
- [Bisakah saya mengunduh file saat saya meluncurkan AWS CloudShell di Console Toolbar?](#)
- [Perangkat lunak apa yang sudah diinstal sebelumnya di lingkungan shell saya?](#)

- [Bisakah saya menginstal perangkat lunak yang tidak tersedia di lingkungan shell?](#)
- [Bisakah saya membatasi tindakan yang dapat dilakukan pengguna? AWS CloudShell](#)
- [Bagaimana saya bisa memindahkan data dari direktori home saya jika saya ingin mengubah Wilayah AWS tempat saya menggunakan AWS CloudShell?](#)
- [Dapatkan saya meningkatkan batas yang menentukan kapan AWS CloudShell waktu habis karena pengguna tidak aktif?](#)
- [Bisakah saya mengakses AWS CloudShellAWS Console Mobile Application dari layar beranda?](#)
- [Bagaimana saya bisa meluncurkan AWS CloudShell di AWS Console Mobile Application?](#)
- [Dapatkan saya menggunakan tombol pengubah di keyboard iOS dan Android saya saat menggunakan AWS CloudShell di? AWS Console Mobile Application](#)
- [Bisakah saya membagi tampilan AWS CloudShell tab menjadi beberapa tab di? AWS Console Mobile Application](#)
- [Dapatkan saya mengakses AWS CloudShell pada Bilah Alat Konsol di perangkat seluler?](#)

Bagaimana cara saya mulai menggunakan AWS CloudShell?

Anda dapat memulai dengan meluncurkan AWS CloudShell dalam beberapa langkah dari AWS Management Console. [Untuk melakukan ini, masuk ke konsol menggunakan kredensial Anda Akun AWS atau IAM di <https://console.aws.amazon.com/console/home>.](#)

Untuk informasi selengkapnya, silakan lihat [Memulai dengan AWS CloudShell](#).

Apa yang perlu saya akses AWS CloudShell?

Karena Anda mengakses AWS CloudShell dari AWS Management Console, Anda harus menjadi pengguna IAM yang dapat memberikan alias atau ID akun yang valid, nama pengguna, dan kata sandi.

Untuk memulai AWS CloudShell di konsol, Anda memerlukan izin IAM yang disediakan oleh kebijakan terlampir. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

Apa yang AWS CloudShell ada diConsole Toolbar?

CloudShell Ikon di kiri bawah AWS Management Console.

Bagaimana cara saya meluncurkan AWS CloudShell diConsole Toolbar?

Anda dapat meluncurkan AWS CloudShell Console Toolbar dengan memilih CloudShell ikon di kiri bawah konsol.

Yang Wilayah AWS CloudShell tersedia di?

Untuk daftar titik akhir layanan yang didukung Wilayah AWS dan terkait, lihat [AWS CloudShell halaman](#) di Referensi Umum Amazon Web

Yang Wilayah AWS ditetapkan jika AWS CloudShell tidak tersedia di Wilayah yang dipilih saat Anda meluncurkan CloudShell diConsole Toolbar?

Wilayah default ditetapkan ke Wilayah yang paling dekat dengan Wilayah yang dipilih. Untuk informasi selengkapnya, lihat [Memilih Wilayah AWS CloudShell, meluncurkan, dan memilih shell](#).

Anda dapat menjalankan perintah yang menyediakan izin untuk mengelola sumber daya di Wilayah yang berbeda dari Region default. Untuk informasi lebih lanjut, lihat [Bekerja di Wilayah AWS](#).

Jenis cangkang apa yang bisa saya gunakan AWS CloudShell?

Di AWS CloudShell, Anda dapat menjalankan perintah menggunakan Bash shell PowerShell, atau Z shell. Untuk mengganti shell, masukkan nama shell yang ingin Anda gunakan menggunakan format berikut pada prompt perintah:

- bash: Gunakan Bash shell
- pwsh: Gunakan PowerShell
- zsh: Gunakan Z shell

Browser web apa yang dapat saya gunakan AWS CloudShell?

AWS CloudShell mendukung versi terbaru dari browser Google Chrome, Mozilla Firefox, Microsoft Edge, dan Apple Safari.

Bagaimana cara membuat dan mengelola AWS CloudShell lingkungan saya?

AWS CloudShell Lingkungan Anda dibuat dan dikelola per ID pengguna IAM per Wilayah. Anda dapat memeriksa UserId dengan menjalankan `aws sts get-caller-identity`. Lingkungan

dimiliki oleh ID pengguna IAM di Wilayah tertentu. Anda akan dapat mengakses AWS CloudShell lingkungan yang berbeda jika Anda mengubah IAM UserId atau Wilayah.

Browser web mana yang dapat saya gunakan ketika saya meluncurkan AWS CloudShell diConsole Toolbar?

Anda dapat meluncurkan CloudShell Console Toolbar menggunakan versi terbaru dari Google Chrome, Microsoft Edge, Mozilla Firefox, dan browser Apple Safari.

Dapatkah saya mengunduh file ketika saya meluncurkan AWS CloudShell diConsole Toolbar?

Ya, Anda dapat mengunduh file saat Anda meluncurkan CloudShell di fileConsole Toolbar. Anda dapat mengunduh file menggunakan versi terbaru Google Chrome dan browser Microsoft Edge.

Saat ini, Anda tidak dapat mengunduh file menggunakan browser Mozilla Firefox dan Apple Safari.

Perangkat lunak apa yang sudah diinstal sebelumnya di lingkungan shell saya?

Dengan shell yang dibuat untuk AWS CloudShell sesi, Anda dapat beralih dengan mulus di antara shell baris perintah pilihan Anda (Bash, PowerShell, dan). Z shell Anda juga dapat memiliki akses ke alat dan utilitas pra-instal seperti Make, pip, sudo, tar, tmuxVim, Wget dan Zip.

Lingkungan shell sudah dikonfigurasi sebelumnya dengan dukungan untuk sebagian besar bahasa perangkat lunak utama. Misalnya, Anda dapat menggunakannya untuk menjalankan Node.js dan Python proyek tanpa harus terlebih dahulu melakukan instalasi runtime. PowerShell pengguna dapat menggunakan .NET Core runtime.

Anda dapat menambahkan file yang dibuat menggunakan shell atau diunggah dengan antarmuka shell ke repositori yang dikendalikan versi yang dikelola menggunakan versi pra-instal. git

Untuk informasi selengkapnya, lihat [Perangkat lunak pra-instal](#).

Bisakah saya menginstal perangkat lunak yang tidak tersedia di lingkungan shell?

Ya, AWS CloudShell pengguna memiliki sudo hak istimewa dan dapat menginstal perangkat lunak dari baris perintah. Untuk informasi selengkapnya, lihat [Menginstal perangkat lunak pihak ketiga di lingkungan shell Anda](#).

Bisakah saya membatasi tindakan yang dapat dilakukan pengguna? AWS CloudShell

Ya, Anda dapat mengontrol tindakan apa yang dapat dilakukan pengguna AWS CloudShell. Misalnya, Anda dapat mengizinkan pengguna untuk mengakses AWS CloudShell tetapi mencegah mereka mengunggah atau mengunduh file dalam lingkungan shell. Atau, sebagai alternatif, Anda dapat sepenuhnya mencegah pengguna mengakses AWS CloudShell. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

Bagaimana saya bisa memindahkan data dari direktori home saya jika saya ingin mengubah Wilayah AWS tempat saya menggunakan AWS CloudShell?

Untuk memindahkan AWS CloudShell data Anda dari satu Wilayah AWS ke Wilayah lain, pertama-tama unduh konten direktori home Anda di satu Wilayah ke mesin lokal Anda, selanjutnya unggah dari itu ke direktori home di Wilayah lain. Untuk informasi selengkapnya, lihat [Tutorial: Menyalin beberapa file antara mesin lokal Anda dan AWS CloudShell](#).

Dapatkah saya meningkatkan batas yang menentukan kapan AWS CloudShell waktu habis karena pengguna tidak aktif?

Sesi shell Anda secara otomatis berakhir setelah sekitar 20-30 menit jika Anda tidak berinteraksi dengan AWS CloudShell menggunakan keyboard atau pointer Anda. Proses yang berjalan tidak dihitung sebagai interaksi. Karena CloudShell dirancang untuk fokus, kegiatan berbasis tugas, tidak ada rencana saat ini untuk meningkatkan [batas waktu tunggu](#) ini.

Jika Anda ingin melakukan tugas berbasis terminal menggunakan batas waktu Layanan AWS yang lebih fleksibel, sebaiknya gunakan IDE berbasis cloud [AWS Cloud9](#), atau meluncurkan dan menyambungkan ke instans [Amazon EC2](#).

Dapatkah saya mengakses AWS CloudShellAWS Console Mobile Application dari layar beranda?

Ya, Anda dapat mengakses AWS CloudShell AWS Console Mobile Application dengan masuk ke Console Mobile Application. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS Console Mobile Application](#).

Bagaimana saya bisa meluncurkan AWS CloudShell di AWS Console Mobile Application?

Anda dapat meluncurkan AWS CloudShell menggunakan salah satu metode berikut:

1. Pilih AWS CloudShellikon di bagian bawah bilah navigasi.
2. Pilih AWS CloudShellpada menu Layanan.

Bisakah saya menggunakan tombol pengubah di keyboard iOS dan Android saya saat menggunakan AWS CloudShell di? AWS Console Mobile Application

Ya, Anda dapat menggunakan tombol pengubah di keyboard iOS dan Android Anda. Untuk informasi selengkapnya, lihat [Panduan Pengguna Aplikasi AWS Console Mobile Application](#).

Bisakah saya membagi tampilan AWS CloudShell tab menjadi beberapa tab di? AWS Console Mobile Application

Tidak, saat ini Anda tidak dapat menjalankan banyak AWS CloudShell tab di aplikasi seluler Anda.

Dapatkah saya mengakses AWS CloudShellConsole Toolbar di perangkat seluler?

Tidak, saat ini Anda tidak dapat mengakses AWS CloudShell Console Toolbar di perangkat seluler Anda.

Memulai dengan AWS CloudShell

Tutorial pengantar ini menunjukkan kepada Anda cara meluncurkan AWS CloudShell dan melakukan tugas-tugas utama menggunakan antarmuka baris perintah shell.

Pertama, Anda masuk ke AWS Management Console dan pilih fileWilayah AWS. Anda kemudian meluncurkan CloudShell di jendela browser baru dan jenis shell untuk bekerja dengan.

Selanjutnya, Anda membuat folder baru di direktori home Anda dan mengunggah file ke sana dari mesin lokal Anda. Anda mengerjakan file itu menggunakan editor pra-instal sebelum menjalankannya sebagai program dari baris perintah. Terakhir, Anda memanggil AWS CLI perintah untuk membuat bucket Amazon S3 dan menambahkan file Anda sebagai objek ke bucket.

Prasyarat

Izin IAM

Anda dapat memperoleh izin AWS CloudShell dengan melampirkan kebijakan AWS terkelola berikut ke identitas IAM Anda (seperti pengguna, peran, atau grup):

- `AWSCloudShellFullAccess`: Menyediakan pengguna dengan akses penuh ke AWS CloudShell dan fitur-fiturnya.

Untuk tutorial ini, Anda juga berinteraksi dengan Layanan AWS. Lebih khusus lagi, Anda berinteraksi dengan Amazon S3 dengan membuat bucket S3 dan menambahkan objek ke bucket itu. Identitas IAM Anda memerlukan kebijakan yang memberikan, setidaknya, izin `s3:CreateBucket` dan `s3:PutObject` izin.

Untuk informasi selengkapnya, lihat [Tindakan Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

File latihan

Latihan ini juga melibatkan mengunggah dan mengedit file yang kemudian dijalankan sebagai program dari antarmuka baris perintah. Buka editor teks di mesin lokal Anda dan tambahkan cuplikan kode berikut.

```
import sys
```

```
x=int(sys.argv[1])
y=int(sys.argv[2])
sum=x+y
print("The sum is",sum)
```

Simpan file dengan nama `add_prog.py`.

Daftar Isi

- [Langkah 1: Masuk ke AWS Management Console](#)
- [Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell](#)
- [Langkah 3: Unduh file dari AWS CloudShell](#)
- [Langkah 4: Unggah file ke AWS CloudShell](#)
- [Langkah 5: Hapus file dari AWS CloudShell](#)
- [Langkah 6: Buat cadangan direktori home](#)
- [Langkah 7: Mulai ulang sesi shell](#)
- [Langkah 8: Hapus direktori home sesi shell](#)
- [Langkah 9: Edit kode file Anda dan jalankan dari baris perintah](#)
- [Langkah 10: Gunakan AWS CLI untuk menambahkan file sebagai objek di bucket Amazon S3](#)

Langkah 1: Masuk ke AWS Management Console

Langkah ini melibatkan memasukkan informasi pengguna IAM Anda untuk mengakses AWS Management Console. Jika Anda sudah berada di konsol, lewati ke [langkah 2](#).

- Anda dapat mengakses AWS Management Console dengan menggunakan URL masuk pengguna IAM atau membuka halaman masuk utama.

IAM user sign-in URL

- Buka browser dan masukkan URL masuk berikut. Ganti `account_alias_or_id` dengan alias akun atau ID akun yang diberikan administrator Anda.

```
https://account_alias_or_id.signin.aws.amazon.com/console/
```

- Masukkan kredensi masuk IAM Anda dan pilih Masuk.

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

Main sign-in page

- Buka <https://aws.amazon.com/console/>.
- Jika sebelumnya Anda tidak masuk menggunakan browser ini, halaman masuk utama akan muncul. Pilih pengguna IAM, masukkan alias akun atau ID akun, dan pilih Berikutnya.

Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Account ID (12 digits) or account alias

Next

- Jika Anda sudah masuk sebagai pengguna IAM sebelumnya. Browser Anda mungkin mengingat alias akun atau ID akun untuk. Akun AWS Jika demikian, masukkan kredensial masuk IAM Anda dan pilih Masuk.

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Sign in

[Sign in using root user email](#)

[Forgot password?](#)


 Note

Anda juga dapat masuk sebagai [pengguna root](#). Identitas ini memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari, bahkan yang administratif. Sebagai gantinya, patuhi praktik terbaik dalam menggunakan pengguna root saja untuk membuat pengguna IAM pertama Anda.


Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell

Pada langkah ini, Anda meluncurkan AWS CloudShell dari antarmuka konsol, memilih yang tersedia Wilayah AWS, dan beralih ke shell pilihan Anda, seperti Bash, PowerShell, atau Z shell.

1. Untuk memilih tempat bekerja, buka menu Pilih Wilayah dan pilih [AWS Wilayah yang didukung](#) untuk bekerja. Wilayah AWS (Wilayah yang Tersedia disorot.)

 Important

Jika Anda beralih Wilayah, antarmuka menyegarkan dan nama yang dipilih Wilayah AWS ditampilkan di atas teks baris perintah. File apa pun yang Anda tambahkan ke penyimpanan persisten hanya tersedia dalam hal yang sama Wilayah AWS. Jika Anda mengubah Wilayah, penyimpanan dan file yang berbeda dapat diakses.

 Important

Jika CloudShell tidak tersedia di Wilayah yang dipilih saat Anda meluncurkan CloudShell di Console Toolbar, di kiri bawah konsol, maka Wilayah default disetel ke Wilayah yang paling dekat dengan Wilayah yang dipilih. Anda dapat menjalankan perintah yang menyediakan izin untuk mengelola sumber daya di Wilayah yang berbeda dari Region default. Untuk informasi lebih lanjut, lihat [Bekerja di Wilayah AWS](#).

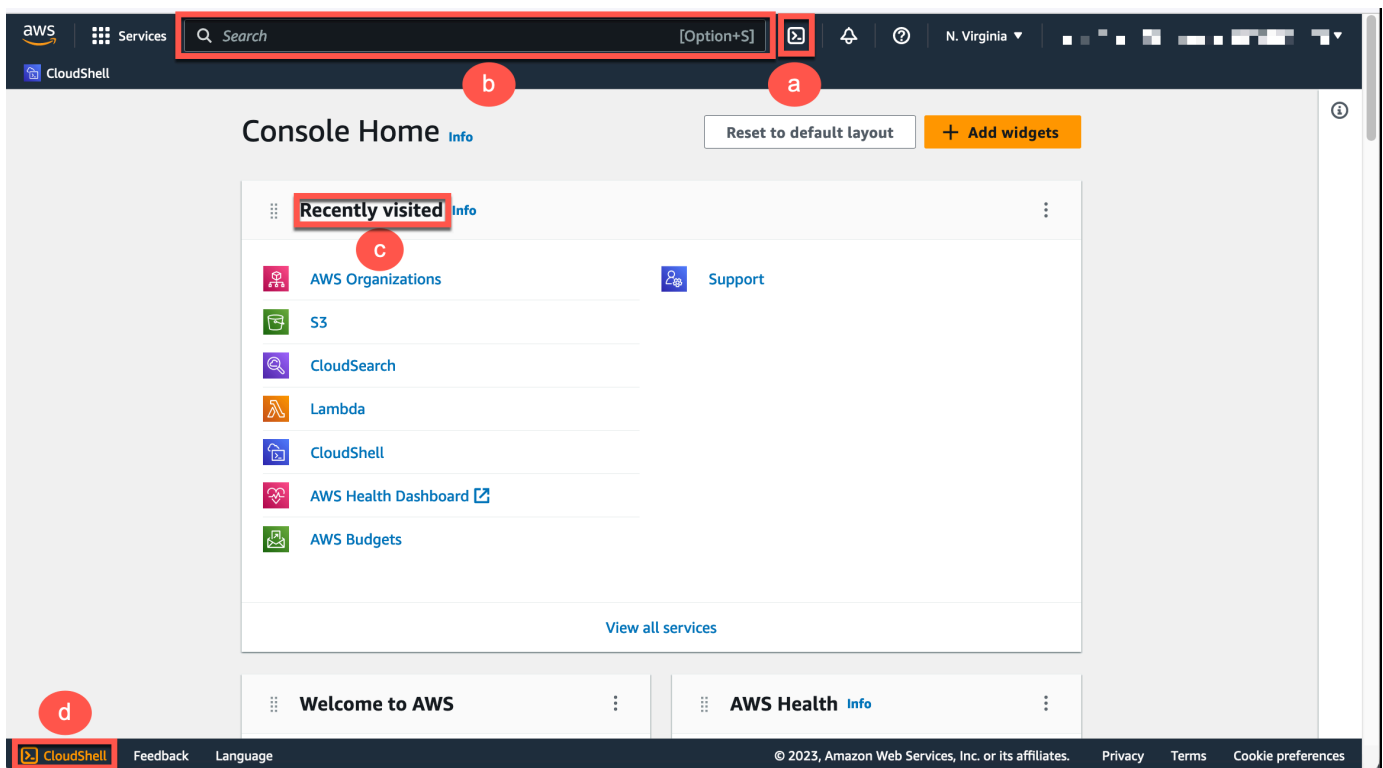
Example

Contoh

Jika Anda memilih Eropa (Spanyol) eu-south-2 tetapi CloudShell tidak tersedia di Eropa (Spanyol)eu-south-2, maka Wilayah default diatur ke Eropa (Irlandia)eu-west-1, yang paling dekat dengan Eropa (Spanyol)eu-south-2.

Anda akan menggunakan kuota layanan untuk Wilayah default, Eropa (Irlandia) eu-west-1 dan CloudShell sesi yang sama akan dipulihkan di semua Wilayah. Wilayah default dapat diubah dan Anda akan diberi tahu di jendela CloudShell browser.

2. Dari AWS Management Console, Anda dapat meluncurkan CloudShell dengan memilih salah satu opsi berikut:
 1. Pada bilah navigasi, pilih CloudShell ikon.
 2. Di kotak Pencarian, ketik "CloudShell", lalu pilih CloudShell.
 3. Di widget yang baru dikunjungi, pilih CloudShell.
 4. Pilih CloudShell pada Console Toolbar, di kiri bawah konsol.
 - Untuk menyesuaikan ketinggian CloudShell sesi Anda, seret=.
 - Untuk mengalihkan CloudShell sesi Anda ke layar penuh, klik Buka di ikon tab browser baru.



Ketika command prompt ditampilkan, shell siap untuk interaksi.

 Note


Jika Anda mengalami masalah yang mencegah Anda berhasil meluncurkan atau berinteraksi AWS CloudShell, periksa informasi untuk mengidentifikasi dan mengatasi masalah tersebut. [Pemecahan Masalah AWS CloudShell](#)

3. Untuk memilih shell pra-instal untuk bekerja dengan, masukkan nama programnya pada prompt baris perintah.

Bash

```
bash
```

Jika Anda beralih ke Bash, simbol pada prompt perintah diperbarui ke \$.

 Note

Bash adalah shell default yang berjalan saat Anda meluncurkan AWS CloudShell.

PowerShell

```
pwsh
```

Jika Anda beralih ke PowerShell, simbol pada prompt perintah diperbarui ke PS>.

Z shell

```
zsh
```

Jika Anda beralih ke Z shell, simbol pada prompt perintah diperbarui ke %.

Untuk informasi tentang versi yang telah diinstal sebelumnya di lingkungan shell Anda, lihat [tabel shell](#) di bagian [lingkungan CloudShell komputasi AWS](#).

Langkah 3: Unduh file dari AWS CloudShell

Langkah ini memandu Anda melalui proses mengunduh file.

1. Untuk mengunduh file, buka Tindakan dan pilih Unduh file dari menu.

Kotak dialog Download file ditampilkan.

2. Di kotak dialog Unduh file, masukkan jalur untuk file yang akan diunduh.

Download file ✕

Download files from your AWS CloudShell to your local desktop. Folders are not supported.

Individual file path
You can copy the file path from the command-line and paste it below.

myfile.txt or /folder/myfile.txt

Cancel Download

Note

Anda dapat menggunakan jalur absolut atau relatif saat menentukan file untuk diunduh. Dengan nama jalur relatif, `/home/cloudshell-user/` ditambahkan secara otomatis ke awal secara default. Jadi, untuk mengunduh file bernama `mydownload-file`, kedua hal berikut ini adalah jalur yang valid:

- Jalur absolut: `/home/cloudshell-user/subfolder/mydownloadfile.txt`
- Jalur relatif: `subfolder/mydownloadfile.txt`

3. Pilih Unduh.

Jika jalur file benar, kotak dialog akan ditampilkan. Anda dapat menggunakan kotak dialog ini untuk membuka file dengan aplikasi default. Atau, Anda dapat menyimpan file ke folder di mesin lokal Anda.

Note

Opsi Unduh tidak tersedia saat Anda meluncurkan CloudShell di fileConsole Toolbar. Anda dapat mengunduh file dari CloudShell konsol atau menggunakan browser web Chrome. Untuk informasi selengkapnya tentang cara mengunduh file, lihat [Langkah 3: Mengunduh file dari file AWS CloudShell](#).

Langkah 4: Unggah file ke AWS CloudShell

Langkah ini menjelaskan cara mengunggah file dan kemudian memindahkannya ke direktori baru di direktori home Anda.

1. Untuk memeriksa direktori kerja Anda saat ini, pada prompt masukkan perintah berikut:

```
pwd
```

Ketika Anda menekan Enter, shell mengembalikan direktori kerja Anda saat ini (misalnya, /home/cloudshell-user).

2. Untuk mengunggah file ke direktori ini, buka Tindakan dan pilih Unggah file dari menu.

Kotak dialog Upload file ditampilkan.

3. Pilih Telusuri.
4. Di kotak dialog File upload sistem Anda, pilih file teks yang Anda buat untuk tutorial ini (add_prog.py) dan pilih Buka.
5. Di kotak dialog Unggah file, pilih Unggah.

Bilah kemajuan melacak unggahan. Jika unggahan berhasil, pesan mengonfirmasi bahwa add_prog.py telah ditambahkan ke root direktori home Anda.

6. Untuk membuat direktori untuk file, masukkan perintah make directory: `mkdir mysub_dir`.
7. Untuk memindahkan file yang diunggah dari root direktori home Anda ke direktori baru, gunakan mv perintah:

```
mv add_prog.py mysub_dir.
```

8. Untuk mengubah direktori kerja Anda ke direktori baru, masukkan `cd mysub_dir`.

Prompt perintah diperbarui untuk menunjukkan bahwa Anda telah mengubah direktori kerja Anda.

9. Untuk melihat isi direktori saat ini `my_sub_dir`, masukkan `ls` perintah.

Isi direktori kerja terdaftar. Ini termasuk file yang baru saja Anda unggah.

Langkah 5: Hapus file dari AWS CloudShell

Langkah ini menjelaskan cara menghapus file dari AWS CloudShell.

1. Untuk menghapus file dari AWS CloudShell, gunakan perintah shell standar seperti `rm` (hapus).

```
rm my-file-for-removal
```

2. Untuk menghapus beberapa file yang memenuhi kriteria tertentu, jalankan `find` perintah.

Contoh berikut menghapus semua file yang menyertakan akhiran “.pdf” dalam nama mereka.

```
find -type f -name '*.pdf' -delete
```

Note

Misalkan Anda berhenti menggunakan AWS CloudShell secara spesifik Wilayah AWS. Kemudian, data yang ada di penyimpanan persisten Wilayah tersebut dihapus secara otomatis setelah periode tertentu. Untuk informasi selengkapnya, lihat [Penyimpanan Persisten](#).

Langkah 6: Buat cadangan direktori home

1. Buat file cadangan

Buat folder sementara di luar direktori home.

```
HOME_BACKUP_DIR=$(mktemp --directory)
```

Anda dapat menggunakan salah satu opsi berikut untuk membuat cadangan:

a. Buat file cadangan menggunakan tar

Untuk membuat file cadangan menggunakan tar, masukkan perintah berikut:

```
tar \
  --create \
  --gzip \
  --verbose \
  --file=${HOME_BACKUP_DIR}/home.tar.gz \
  [--exclude ${HOME}/.cache] \ // Optional
  ${HOME}/
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.tar.gz"
```

b. Buat file cadangan menggunakan zip

Untuk membuat file cadangan menggunakan zip, masukkan perintah berikut:

```
zip \
  --recurse-paths \
  ${HOME_BACKUP_DIR}/home.zip \
  ${HOME} \
  [--exclude ${HOME}/.cache/\^*] // Optional
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.zip"
```

2. Transfer file cadangan di luar CloudShell

Anda dapat menggunakan salah satu opsi berikut untuk mentransfer file cadangan di luar CloudShell:

a. Unduh file cadangan di mesin lokal Anda

Anda dapat mengunduh file yang dibuat pada langkah sebelumnya. Untuk informasi selengkapnya tentang cara mengunduh file CloudShell, lihat [Mengunduh file dari AWS CloudShell](#).

Di kotak dialog file unduhan, masukkan jalur untuk file yang akan diunduh (misalnya, /tmp/tmp.iA99tD9L98/home.tar.gz).


b. Transfer file cadangan ke S3

Untuk menghasilkan bucket, masukkan perintah berikut:

```
aws s3 mb s3://${BUCKET_NAME}
```

Gunakan AWS CLI untuk menyalin file ke bucket S3:

```
aws s3 cp ${HOME_BACKUP_DIR}/home.tar.gz s3://${BUCKET_NAME}
```

 Note


Biaya transfer data mungkin berlaku.

3. Backup langsung ke bucket S3

Untuk membuat cadangan langsung ke bucket S3, masukkan perintah berikut:

```
aws s3 cp \  
  ${HOME}/ \  
  s3://${BUCKET_NAME} \  
  --recursive \  
  [--exclude .cache/\*] // Optional
```

Langkah 7: Mulai ulang sesi shell

 Note


Sebagai tindakan pengamanan, jika Anda tidak berinteraksi dengan shell menggunakan keyboard atau pointer untuk waktu yang lama, sesi berhenti secara otomatis. Sesi yang berjalan lama juga dihentikan secara otomatis. Untuk informasi selengkapnya, lihat [Sesi Shell](#).

1. Untuk memulai ulang sesi shell, pilih Actions, Restart AWS CloudShell.

Anda diberi tahu bahwa memulai ulang AWS CloudShell menghentikan semua sesi aktif saat ini.
Wilayah AWS


2. Untuk mengonfirmasi, pilih Restart.

Antarmuka menampilkan pesan bahwa lingkungan CloudShell komputasi berhenti. Setelah lingkungan berhenti dan dimulai ulang, Anda dapat mulai bekerja dengan baris perintah di sesi baru.

 Note

Dalam beberapa kasus, mungkin perlu beberapa menit bagi lingkungan Anda untuk memulai kembali.

Langkah 8: Hapus direktori home sesi shell

 Warning

Menghapus direktori home Anda adalah tindakan yang tidak dapat diubah di mana semua data yang disimpan di direktori home Anda dihapus secara permanen. Namun, Anda mungkin ingin mempertimbangkan opsi ini dalam situasi berikut:

- Anda salah memodifikasi file dan tidak dapat mengakses lingkungan AWS CloudShell komputasi. Menghapus direktori home Anda akan kembali AWS CloudShell ke pengaturan defaultnya.
- Anda ingin AWS CloudShell segera menghapus semua data Anda. Jika Anda berhenti menggunakan AWS CloudShell di AWS Wilayah, penyimpanan persisten secara [otomatis dihapus pada akhir periode penyimpanan](#) kecuali Anda meluncurkan AWS CloudShell lagi di Wilayah.

Jika Anda memerlukan penyimpanan jangka panjang untuk file Anda, pertimbangkan layanan seperti Amazon S3 atau CodeCommit

1. Untuk menghapus sesi shell, pilih Actions, Delete AWS CloudShell home directory.

Anda diberi tahu bahwa menghapus direktori AWS CloudShell home akan menghapus semua data yang saat ini disimpan di lingkungan Anda. AWS CloudShell

Note

Anda tidak dapat membatalkan tindakan ini.

2. Untuk mengonfirmasi penghapusan, masukkan hapus di bidang input teks, lalu pilih Hapus.

Delete AWS CloudShell home directory

Deleting your home directory will delete all data currently stored in your AWS CloudShell environment. This action cannot be undone. AWS CloudShell stops all active sessions in the current AWS Region and creates a new environment immediately.

To confirm deletion, enter **delete** in the text input field.

Cancel

Delete

AWS CloudShell menghentikan semua sesi aktif saat ini Wilayah AWS dan segera menciptakan lingkungan baru.

Keluar dari sesi shell secara manual

Dengan baris perintah, Anda dapat meninggalkan sesi shell dan keluar menggunakan `exit` perintah. Anda kemudian dapat menekan tombol apa saja untuk menyambung kembali dan terus menggunakan AWS CloudShell.

Langkah 9: Edit kode file Anda dan jalankan menggunakan baris perintah

Langkah ini menunjukkan cara menggunakan Vim editor pra-instal untuk bekerja dengan file. Anda kemudian menjalankan file itu sebagai program dari baris perintah.

1. Untuk mengedit file yang Anda unggah pada langkah sebelumnya, masukkan perintah berikut:

```
vim add_prog.py
```

Antarmuka shell menyegarkan untuk menampilkan editor. Vim

- Untuk mengedit fileVim, tekan I tombol. Sekarang edit isinya sehingga program menambahkan tiga angka, bukan dua.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
sum=x+y+z
print("The sum is",sum)
```

Note

Jika Anda menempelkan teks ke editor dan mengaktifkan [fitur Tempel Aman](#), peringatan akan ditampilkan. Teks multiline yang disalin dapat berisi skrip berbahaya. Dengan fitur Safe Paste, Anda dapat memverifikasi teks lengkap sebelum ditempelkan. Jika Anda puas bahwa teksnya aman, pilih Tempel.

- Setelah Anda mengedit program, tekan Esc untuk masuk ke mode Vim perintah. Kemudian, masukkan :wq perintah untuk menyimpan file dan keluar dari editor.

Note

Jika Anda baru mengenal mode Vim perintah, Anda mungkin awalnya merasa sulit untuk beralih antara mode perintah dan mode sisipkan. Mode perintah digunakan saat menyimpan file dan keluar dari aplikasi. Mode sisipkan digunakan saat memasukkan teks baru. Untuk masuk ke mode insert, tekanI, dan, untuk masuk ke mode perintah, tekanEsc. Untuk informasi selengkapnya tentang Vim dan alat lain yang tersedia diAWS CloudShell, lihat[Alat pengembangan dan utilitas shell](#).

- Pada antarmuka baris perintah utama, jalankan program berikut dan tentukan tiga angka untuk input. Sintaksnya adalah sebagai berikut.

```
python3 add_prog.py 4 5 6
```

Baris perintah menampilkan output program:The sum is 15.

Langkah 10: Gunakan AWS CLI untuk menambahkan file sebagai objek di bucket Amazon S3

Pada langkah ini, Anda membuat bucket Amazon S3 dan kemudian menggunakan `PutObject` metode untuk menambahkan file kode Anda sebagai objek di bucket itu.

Note

Dalam kebanyakan kasus, Anda dapat [menggunakan layanan seperti CodeCommit](#) untuk memasukkan file perangkat lunak ke dalam repositori yang dikendalikan versi. Tutorial ini menunjukkan bagaimana Anda dapat menggunakannya AWS CLI AWS CloudShell untuk berinteraksi dengan layanan AWS lainnya. Dengan menggunakan metode ini, Anda tidak perlu mengunduh atau menginstal sumber daya tambahan apa pun. Selain itu, karena Anda sudah diautentikasi di dalam shell, Anda tidak perlu mengonfigurasi kredensial sebelum melakukan panggilan.

1. Untuk membuat bucket dalam yang ditentukan Wilayah AWS, masukkan perintah berikut:

```
aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
```

Note

Jika Anda membuat bucket di luar `us-east-1` Region, tambahkan `create-bucket-configuration LocationConstraint` parameter untuk menentukan Region. Berikut ini adalah contoh sintaks.

```
$ aws s3api create-bucket --bucket my-bucket --region eu-west-1 --create-bucket-configuration LocationConstraint=eu-west-1
```

Jika panggilan berhasil, baris perintah menampilkan respons dari layanan yang mirip dengan output berikut.

```
{  
  "Location": "/insert-unique-bucket-name-here"
```

```
}
```

Note

Jika Anda tidak mematuhi [aturan penamaan bucket](#), kesalahan berikut akan ditampilkan: Terjadi kesalahan (InvalidBucketName) saat memanggil CreateBucket operasi: Bucket yang ditentukan tidak valid.

2. Untuk mengunggah file dan menambahkan file sebagai objek ke bucket yang baru saja Anda buat, panggil PutObject metode.

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body  
add_prog.py
```

Setelah objek diunggah ke bucket Amazon S3, baris perintah menampilkan respons dari layanan yang mirip dengan output berikut:

```
{"ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebee56\""}
```

ETag itu adalah hash dari objek yang disimpan. Anda dapat menggunakan hash ini untuk [memeriksa integritas objek yang diunggah ke Amazon S3](#).

Topik terkait

- [Bekerja dengan AWS layanan di AWS CloudShell](#)
- [Tutorial: Menyalin beberapa file antara mesin lokal Anda dan AWS CloudShell](#)
- [Tutorial: Menggunakan CodeCommit di AWS CloudShell](#)
- [Bekerja dengan AWS CloudShell](#)
- [Menyesuaikan Anda AWS CloudShell pengalaman](#)

Tutorial AWS CloudShell

Tutorial berikut akan memungkinkan Anda untuk bereksperimen dan menguji berbagai fungsi dan integrasi saat menggunakan AWS CloudShell

Topik

- [Tutorial: Menyalin beberapa file antara mesin lokal Anda dan AWS CloudShell](#)
- [Tutorial: Menggunakan CodeCommit di AWS CloudShell](#)
- [Tutorial: Membuat URL yang telah ditandatangani sebelumnya untuk objek Amazon S3 menggunakan AWS CloudShell](#)
- [Tutorial: Membangun wadah Docker di dalam AWS CloudShell dan mendorongnya ke repositori Amazon ECR](#)
- [Tutorial: Menyebarkan fungsi Lambda menggunakan AWS CDK](#)

Tutorial: Menyalin beberapa file antara mesin lokal Anda dan AWS CloudShell

Dengan menggunakan CloudShell antarmuka, Anda dapat mengunggah atau mengunduh satu file antara mesin lokal Anda dan lingkungan shell sekaligus. Untuk menyalin beberapa file antara CloudShell dan komputer lokal Anda secara bersamaan, gunakan salah satu opsi berikut:

- Amazon S3: Gunakan bucket S3 sebagai perantara saat menyalin file antara mesin lokal Anda dan CloudShell.
- File zip: Kompres beberapa file dalam satu folder zip yang dapat diunggah atau diunduh menggunakan CloudShell antarmuka.

Note

Karena CloudShell tidak mengizinkan lalu lintas internet masuk, saat ini tidak mungkin untuk menggunakan perintah seperti `scp` atau `rsync` menyalin beberapa file antara mesin lokal dan lingkungan CloudShell komputasi.

Mengunggah dan mengunduh beberapa file menggunakan Amazon S3

Prasyarat

Untuk bekerja dengan bucket dan objek, Anda memerlukan kebijakan IAM yang memberikan izin untuk melakukan tindakan API Amazon S3 berikut:

- `s3:CreateBucket`
- `s3:PutObject`
- `s3:GetObject`

Untuk daftar lengkap tindakan Amazon S3, lihat [Tindakan](#) di Referensi API Amazon Simple Storage Service.

Unggah beberapa file untuk AWS CloudShell menggunakan Amazon S3

1. Di AWS CloudShell, buat bucket S3 dengan menjalankans3 perintah berikut:

```
aws s3api create-bucket --bucket your-bucket-name --region us-east-1
```

Jika panggilan berhasil, baris perintah menampilkan respons dari layanan S3:

```
{
  "Location": "/your-bucket-name"
}
```

2. Unggah file dalam direktori dari mesin lokal Anda ke bucket. Pilih salah satu opsi berikut untuk mengunggah file:
 - AWS Management Console: Gunakan drag-and-drop untuk mengunggah file dan folder ke bucket.
 - AWS CLI: Dengan versi alat yang diinstal pada mesin lokal Anda, gunakan baris perintah untuk mengunggah file dan folder ke bucket.

Using the console

- Buka konsol Amazon S3 di <https://s3.console.aws.amazon.com/s3/>.

(Jika Anda menggunakan AWS CloudShell, Anda harus sudah login ke konsol.)

- Di panel navigasi kiri, pilih Bucket, lalu pilih nama bucket yang ingin Anda unggah folder atau file Anda. Anda juga dapat membuat bucket pilihan Anda dengan memilih Create bucket.
- Untuk memilih file dan folder yang ingin Anda unggah, pilih Unggah. Lalu tarik dan letakkan file dan folder yang Anda pilih ke dalam jendela konsol yang mencantumkan objek dalam bucket tujuan, atau pilih Tambahkan file, atau Tambahkan folder.

File yang Anda pilih tercantum di Unggah yang baru.

- Pilih kotak centang untuk menunjukkan file yang akan ditambahkan.
- Untuk menambahkan file yang dipilih ke bucket, pilih Unggah.

Note

Untuk informasi lebih lanjut tentang opsi konfigurasi lengkap saat menggunakan konsol, lihat [Bagaimana cara mengunggah file dan folder ke bucket S3?](#) dalam Panduan Pengguna Amazon Simple Storage Service.

Using AWS CLI

Note

Untuk opsi ini, Anda harus memiliki AWS CLI alat yang diinstal pada komputer lokal Anda dan memiliki kredensi Anda dikonfigurasi untuk panggilan ke AWS layanan. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS Command Line Interface](#).


- Luncurkan AWS CLI alat dan jalankan `aws s3` perintah berikut untuk menyinkronkan bucket yang ditentukan dengan isi direktori saat ini di komputer lokal Anda:

```
aws s3 sync folder-path s3://your-bucket-name
```

Jika sinkronisasi berhasil, pesan upload akan ditampilkan untuk setiap objek yang ditambahkan ke bucket.

3. Kembali ke baris CloudShell perintah dan masukkan perintah berikut untuk menyinkronkan direktori di lingkungan shell dengan isi bucket S3:


```
aws s3 sync s3://your-bucket-name folder-path
```

 Note

Anda juga dapat menambahkan `--exclude "<value>"` dan `--include "<value>"` parameter kesync perintah untuk melakukan pencocokan pola baik mengecualikan atau menyertakan file atau objek tertentu.

Untuk informasi lebih lanjut, lihat [Penggunaan Kecualikan dan Sertakan Filter](#) dalam ReferensiAWS CLI Perintah.

Jika sinkronisasi berhasil, pesan unduhan ditampilkan untuk setiap file yang diunduh dari bucket ke direktori.

 Note

Dengan perintah sinkronisasi, hanya file baru dan yang diperbarui yang disalin secara rekursif dari direktori sumber ke tujuan.

Unduh beberapa file dariAWS CloudShell menggunakan Amazon S3

1. Dengan menggunakan barisAWS CloudShell perintah, masukkan`aws s3` perintah berikut untuk menyinkronkan bucket S3 dengan isi direktori saat ini di lingkungan shell:

```
aws s3 sync folder-path s3://your-bucket-name
```


Note

Anda juga dapat menambahkan `--exclude "<value>"` dan `--include "<value>"` parameter `kesync` perintah untuk melakukan pencocokan pola baik mengecualikan atau menyertakan file atau objek tertentu.

Untuk informasi lebih lanjut, lihat [Penggunaan Kecualikan dan Sertakan Filter](#) dalam Referensi AWS CLI Perintah.

Jika sinkronisasi berhasil, pesan upload akan ditampilkan untuk setiap objek yang ditambahkan ke bucket.

2. Unduh konten bucket ke mesin lokal Anda. Karena konsol Amazon S3 tidak mendukung pengunduhan beberapa objek, Anda perlu menggunakan AWS CLI alat yang diinstal pada mesin lokal Anda.

Dari baris perintah AWS CLI alat, jalankan perintah berikut:

```
aws s3 sync s3://your-bucket-name folder-path
```

Jika sinkronisasi berhasil, baris perintah menampilkan pesan unduhan untuk setiap file yang diperbarui atau ditambahkan di direktori tujuan.

Note

Untuk opsi ini, Anda harus memiliki AWS CLI alat yang diinstal pada komputer lokal Anda dan memiliki kredensi Anda dikonfigurasi untuk panggilan ke AWS layanan. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS Command Line Interface](#).

Mengunggah dan mengunduh beberapa file menggunakan folder zip

Dengan utilitas `zip/unzip`, Anda dapat mengompres beberapa file dalam arsip yang dapat diperlakukan sebagai satu file. Utilitas sudah diinstal sebelumnya di lingkungan CloudShell komputasi.

Untuk informasi lebih lanjut tentang alat pra-instal, lihat [Alat pengembangan dan utilitas shell](#).

Unggah beberapa file untuk AWS CloudShell menggunakan folder zip

1. Di komputer lokal Anda, tambahkan file yang akan diunggah ke folder zip.
2. Luncurkan CloudShell, lalu pilih Tindakan, Unggah file.
3. Dalam Unggah file kotak dialog, pilih Pilih file, lalu pilih folder zip yang baru saja Anda buat.
4. Dalam Upload file kotak dialog, pilih Upload untuk menambahkan file yang dipilih ke lingkungan shell.
5. Pada baris CloudShell perintah, jalankan perintah berikut untuk unzip isi arsip zip ke direktori tertentu:

```
unzip zipped-files.zip -d my-unzipped-folder
```

Unduh beberapa file dari AWS CloudShell menggunakan folder zip

1. Di baris CloudShell perintah, jalankan perintah berikut untuk menambahkan semua file dalam direktori saat ini ke folder zip:

```
zip -r zipped-archive.zip *
```

2. Pilih Tindakan, Unduh file.
3. Di kotak dialog Download file, masukkan path untuk folder zip (`/home/cloudshell-user/zip-folder/zipped-archive.zip`, misalnya), dan kemudian pilih Download.

Jika jalurnya benar, dialog browser menawarkan pilihan untuk membuka folder zip atau menyimpannya ke mesin lokal Anda.

4. Pada komputer lokal Anda, Anda sekarang dapat unzip isi folder zip download.

Tutorial: Menggunakan CodeCommit di AWS CloudShell

CodeCommit adalah layanan kontrol sumber terkelola yang aman, dan sangat terukur yang menghost repositori Git privat. Menggunakan AWS CloudShell, Anda dapat bekerja dengan CodeCommit pada baris perintah menggunakan `git-remote-codecommit` utilitas. Utilitas ini sudah terinstal di lingkungan AWS CloudShell komputasi dan menyediakan metode sederhana untuk mendorong dan menarik kode dari CodeCommit repositori. Utilitas ini melakukan ini dengan memperluas Git. Untuk informasi selengkapnya, lihat [Panduan Pengguna AWS CodeCommit](#).

Tutorial ini menjelaskan cara membuat sebuah CodeCommit repositori dan mengklon ke lingkungan AWS CloudShell komputasi Anda. Anda juga mempelajari cara melakukan stage dan melakukan commit ke repositori kloning Anda sebelum mendorongnya ke repositori jarak jauh yang dikelola di AWS Cloud.

Prasyarat

Untuk informasi tentang izin yang diperlukan pengguna IAM untuk menggunakan AWS CloudShell, lihat [bagian prasyarat dalam tutorial Memulai](#). Anda juga memerlukan [izin IAM](#) untuk bekerja sama CodeCommit.

Selain itu, sebelum memulai, pastikan untuk memiliki yang berikut:

- Pemahaman dasar tentang perintah Git dan konsep kontrol versi
- Sebuah file di direktori home shell Anda yang dapat berkomitmen untuk repositori lokal dan remote. Dalam tutorial ini, ini disebut sebagai `my-git-file`.

Langkah 1: Membuat dan mengklon CodeCommit repositori

1. Di antarmuka baris CloudShell perintah, masukkan `codecommit` perintah berikut untuk membuat CodeCommit repositori yang disebut `MyDemoRepo`.

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-  
description "My demonstration repository"
```

Jika repositori berhasil dibuat, baris perintah menampilkan respons layanan.

```
{  
  "repositoryMetadata": {  
    "accountId": "111122223333",  
    "repositoryId": "0dcd29a8-941a-1111-1111-11111111111a",  
    "repositoryName": "MyDemoRepo",  
    "repositoryDescription": "My demonstration repository",  
    "lastModifiedDate": "2020-11-23T20:38:23.068000+00:00",  
    "creationDate": "2020-11-23T20:38:23.068000+00:00",  
    "cloneUrlHttp": "https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/  
MyDemoRepo",  
    "cloneUrlSsh": "ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/  
MyDemoRepo",  
    "Arn": "arn:aws:codecommit:eu-west-1:111111111111:MyDemoRepo"
```

```
}  
)
```

2. Dengan menggunakan baris perintah, buat direktori baru untuk repositori lokal Anda dan jadikan direktori kerja Anda.

```
mkdir my-shell-repo  
cd my-shell-repo
```

3. Untuk mengklon repositori jarak jauh, gunakan `git clone` perintah. (Saat Anda bekerja sama dengan `git-remote-codecommit`, gunakan gaya URL HTTPS (GRC)).

```
git clone codecommit::eu-west-1://MyDemoRepo
```

Jika repositori berhasil dikloning, baris perintah menampilkan respons layanan.

```
Cloning into 'MyDemoRepo'...  
warning: You appear to have cloned an empty repository.
```

4. Untuk menavigasi ke repositori kloning, gunakan `cd` perintah.

```
cd MyDemoRepo
```

Langkah 2: Tahap dan komit file sebelum mendorongnya ke CodeCommit repositori Anda

1. Tambahkan file yang dipanggil `my-git-file` ke `MyDemoRepo` folder menggunakan editor Vim atau fitur unggah file AWS CloudShell. Untuk mempelajari cara menggunakan keduanya, lihat [tutorial Memulai](#).
2. Untuk tahap file Anda di repositori, jalankan `add` perintah git.

```
git add my-git-file
```

3. Untuk memeriksa apakah file telah dipentaskan dan siap untuk dilakukan, jalankan `status` perintah git.

```
git status
```

`my-git-file` terdaftar sebagai file baru dan ditampilkan dalam teks hijau, menunjukkan itu siap untuk berkomitmen.

4. Komit versi file bertahap ini ke repositori.

```
git commit -m "first commit to repo"
```

Note

Jika Anda diminta untuk informasi konfigurasi untuk menyelesaikan komit, gunakan format berikut.

```
$ git config --global user.name "Jane Doe"  
$ git config --global user.email janedoe@example.com
```

5. Untuk menyinkronkan repositori jarak jauh Anda dengan perubahan yang dibuat di lokal Anda, dorong perubahan ke cabang upstream.

```
git push
```

Tutorial: Membuat URL yang telah ditandatangani sebelumnya untuk objek Amazon S3 menggunakan AWS CloudShell

Tutorial ini menampilkan cara membuat URL untuk berbagi objek Amazon S3 dengan orang lain. Karena pemilik objek menentukan kredensi keamanan mereka sendiri saat berbagi, siapa pun yang menerima URL yang telah ditandatangani sebelumnya dapat mengakses objek untuk waktu yang terbatas.

Prasyarat

- Pengguna IAM dengan izin akses yang disediakan oleh `AWSCloudShellFullAccess` kebijakan.
- Untuk izin IAM yang diperlukan untuk membuat URL yang ditandatangani sebelumnya, lihat [Membagikan objek dengan orang lain](#) di Panduan Pengguna Amazon Simple Storage Service.

Langkah 1: Buat peran IAM untuk memberikan akses ke bucket Amazon S3

1. Untuk mendapatkan detail IAM Anda yang dapat dibagikan, hubungiget-caller-identity perintah dariAWS CloudShell.

```
aws sts get-caller-identity
```

Jika panggilan berhasil, baris perintah menampilkan respon.

```
{
  "Account": "123456789012",
  "UserId": "AROAXX0ZUU0TTWDCVIDZ2:redirect_session",
  "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
}
```

2. Ambil informasi pengguna yang Anda peroleh pada langkah sebelumnya, dan tambahkan keAWS CloudFormation template. Template ini membuat peran IAM. Peran ini memberikan izin hak istimewa kepada kolaborator Anda untuk sumber daya bersama.

```
Resources:
  CollaboratorRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              AWS: "arn:aws:iam::531421766567:role/Feder08"
            Action: "sts:AssumeRole"
      Description: Role used by my collaborators
      MaxSessionDuration: 7200
  CollaboratorPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 's3:*'
```

```

    Resource: 'arn:aws:s3:::<YOUR_BUCKET_FOR_FILE_TRANSFER>'
    Condition:
      StringEquals:
        s3:prefix:
          - "myfolder/*"
    PolicyName: S3ReadSpecificFolder
    Roles:
      - !Ref CollaboratorRole
Outputs:
  CollaboratorRoleArn:
    Description: Arn for the Collaborator's Role
    Value: !GetAtt CollaboratorRole.Arn

```

3. Simpan AWS CloudFormation template dalam file yang bernama `template.yaml`.
4. Gunakan template untuk menyebarkan tumpukan dan membuat peran IAM dengan memanggil `deploy` perintah.

```

aws cloudformation deploy --template-file ./template.yaml --stack-name
  CollaboratorRole --capabilities CAPABILITY_IAM

```

Buat URL

1. Menggunakan editor Anda di AWS CloudShell, tambahkan kode berikut. Kode ini membuat URL yang menyediakan pengguna federasi dengan akses langsung ke AWS Management Console.

```

import urllib, json, sys
import requests
import boto3
import os

def main():
    sts_client = boto3.client('sts')
    assume_role_response = sts_client.assume_role(
        RoleArn=os.environ.get('ROLE_ARN'),
        RoleSessionName="collaborator-session"
    )
    credentials = assume_role_response['Credentials']
    url_credentials = {}
    url_credentials['sessionId'] = credentials.get('AccessKeyId')
    url_credentials['sessionKey'] = credentials.get('SecretAccessKey')
    url_credentials['sessionToken'] = credentials.get('SessionToken')

```

```

json_string_with_temp_credentials = json.dumps(url_credentials)
print(f"json string {json_string_with_temp_credentials}")

request_parameters = f"?
Action=getSignInToken&Session={urllib.parse.quote(json_string_with_temp_credentials)}"
request_url = "https://signin.aws.amazon.com/federation" + request_parameters
r = requests.get(request_url)
signin_token = json.loads(r.text)
request_parameters = "?Action=login"
request_parameters += "&Issuer=Example.org"
request_parameters += "&Destination=" + urllib.parse.quote("https://us-
west-2.console.aws.amazon.com/cloudshell")
request_parameters += "&SignInToken=" + signin_token["SignInToken"]
request_url = "https://signin.aws.amazon.com/federation" + request_parameters

# Send final URL to stdout
print (request_url)

if __name__ == "__main__":
    main()

```

2. Simpan kode dalam sebuah file bernama `share.py`.
3. Jalankan hal berikut ini dari baris untuk mengambil AWS CloudFormation. Kemudian, gunakan dalam Python skrip untuk mendapatkan kredensial keamanan sementara.

```

ROLE_ARN=$(aws cloudformation describe-stacks --stack-name CollaboratorRole --query
"Stacks[*].Outputs[?OutputKey=='CollaboratorRoleArn'].OutputValue" --output text)
python3 ./share.py

```

Skrip mengembalikan URL yang dapat diklik oleh kolaborator untuk membawanya AWS CloudShell masuk AWS Management Console. Kolaborator memiliki kontrol penuh atas `myfolder/` folder di bucket Amazon S3 selama 3.600 detik berikutnya (1 jam). Kredensial kedaluwarsa setelah satu jam. Setelah waktu ini, kolaborator tidak dapat lagi mengakses bucket.

Tutorial: Membangun wadah Docker di dalam AWS CloudShell dan mendorongnya ke repositori Amazon ECR

Tutorial ini menunjukkan cara mendefinisikan dan membangun wadah Docker AWS CloudShell dan mendorongnya ke repositori Amazon ECR.

Prasyarat

- Anda harus memiliki izin yang diperlukan untuk membuat dan mendorong ke repositori Amazon ECR. Untuk informasi selengkapnya tentang repositori dengan Amazon ECR, lihat [repositori pribadi Amazon ECR](#) di Panduan Pengguna Amazon ECR. Untuk informasi selengkapnya tentang izin yang diperlukan untuk mendorong gambar dengan Amazon ECR, lihat [Izin IAM yang diperlukan untuk mendorong gambar di Panduan Pengguna](#) Amazon ECR.

Prosedur Tutorial

Tutorial berikut menguraikan cara menggunakan CloudShell antarmuka untuk membangun wadah Docker dan mendorongnya ke repositori Amazon ECR.

1. Buat folder baru di direktori home Anda.

```
mkdir ~/docker-cli-tutorial
```

2. Arahkan ke folder yang Anda buat.

```
cd ~/docker-cli-tutorial
```

3. Buat Dockerfile kosong.

```
touch Dockerfile
```

4. Menggunakan editor teks, misalnanano Dockerfile, buka file dan tempel konten berikut ke dalamnya.

```
# Dockerfile

# Base this container on the latest Amazon Linux version
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install the cowsay binary
RUN dnf install --assumeyes cowsay

# Default entrypoint binary
ENTRYPOINT [ "cowsay" ]

# Default argument for the cowsay entrypoint
```

```
CMD [ "Hello, World!" ]
```

5. Dockerfile sekarang siap dibangun. Bangun wadah dengan menjalankandocker build. Tandai wadah dengan easy-to-type nama untuk digunakan dalam perintah future.

```
docker build --tag test-container .
```

Pastikan untuk memasukkan trailing period (.).

6. Anda sekarang dapat menguji wadah untuk memeriksa apakah itu berjalan dengan benar diAWS CloudShell.

```
docker container run test-container
```

7. Sekarang setelah Anda memiliki wadah Docker yang berfungsi, Anda perlu mendorongnya ke repositori Amazon ECR. Jika Anda memiliki repositori Amazon ECR yang ada, Anda dapat melewati langkah ini.

Jalankan perintah berikut untuk membuat repositori Amazon ECR untuk tutorial ini.

```
ECR_REPO_NAME=docker-tutorial-repo  
aws ecr create-repository --repository-name ${ECR_REPO_NAME}
```

8. Setelah Anda membuat repositori Amazon ECR, Anda dapat mendorong wadah Docker ke sana.

Jalankan perintah berikut untuk mendapatkan kredensyal masuk Amazon ECR untuk Docker.

```
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)  
ECR_URL=${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_REGION}.amazonaws.com  
aws ecr get-login-password | docker login --username AWS --password-stdin  
${ECR_URL}
```

9. Tandai gambar dengan repositori Amazon ECR target dan kemudian dorong ke repositori itu.

```
docker tag test-container ${ECR_URL}/${ECR_REPO_NAME}  
docker push ${ECR_URL}/${ECR_REPO_NAME}
```

Jika Anda mengalami kesalahan atau mengalami masalah saat mencoba menyelesaikan tutorial ini, lihat bagian [Pemecahan Masalah](#) dari panduan ini untuk mendapatkan bantuan.

Bersihkan

Anda sekarang telah berhasil menerapkan wadah Docker Anda ke repositori Amazon ECR Anda. Untuk menghapus file yang Anda buat dalam tutorial ini dari AWS CloudShell lingkungan Anda, jalankan perintah berikut.

- ```
cd ~
rm -rf ~/docker-cli-tutorial
```

- Hapus repositori Amazon ECR.

```
aws ecr delete-repository --force --repository-name ${ECR_REPO_NAME}
```

## Tutorial: Menyebarkan fungsi Lambda menggunakan AWS CDK

Tutorial ini menunjukkan cara menerapkan fungsi Lambda ke akun Anda menggunakan AWS Cloud Development Kit (AWS CDK)

### Prasyarat

- Bootstrap akun Anda untuk digunakan dengan file AWS CDK. Untuk informasi tentang bootstrap dengan AWS CDK, lihat [Bootstrapping di Panduan Pengembang v2](#). AWS CDK Jika Anda belum melakukan bootstrap akun, Anda dapat menjalankannya. `cdk bootstrap` CloudShell
- Pastikan Anda memiliki izin yang sesuai untuk menyebarkan sumber daya ke akun Anda. Izin administrator direkomendasikan.

### Prosedur Tutorial

Tutorial berikut menguraikan cara menerapkan fungsi Lambda berbasis kontainer Docker menggunakan AWS CDK

1. Buat folder baru di direktori home Anda.

```
mkdir ~/docker-cdk-tutorial
```

2. Arahkan ke folder yang Anda buat.

```
cd ~/docker-cdk-tutorial
```

### 3. Instal AWS CDK dependensi secara lokal.

```
npm install aws-cdk aws-cdk-lib
```

### 4. Buat AWS CDK proyek kerangka di folder yang Anda buat.

```
touch cdk.json
mkdir lib
touch lib/docker-tutorial.js lib/Dockerfile lib/hello.js
```

### 5. Menggunakan editor teks, misal nyanano cdk.json, buka file dan tempel konten berikut ke dalamnya.

```
{
 "app": "node lib/docker-tutorial.js"
}
```

### 6. Buka lib/docker-tutorial.js file dan tempel konten berikut ke dalamnya.

```
// this file defines the CDK constructs we want to deploy

const { App, Stack } = require('aws-cdk-lib');
const { DockerImageFunction, DockerImageCode } = require('aws-cdk-lib/aws-lambda');
const path = require('path');

// create an application
const app = new App();

// define stack
class DockerTutorialStack extends Stack {
 constructor(scope, id, props) {
 super(scope, id, props);

 // define lambda that uses a Docker container
 const dockerfileDir = path.join(__dirname);
 new DockerImageFunction(this, 'DockerTutorialFunction', {
 code: DockerImageCode.fromImageAsset(dockerfileDir),
 functionName: 'DockerTutorialFunction',
 });
 }
}
```

```
}
}

// instantiate stack
new DockerTutorialStack(app, 'DockerTutorialStack');
```

7. Buka `lib/Dockerfile` dan tempel konten berikut ke dalamnya.

```
Use a NodeJS 20.x runtime
FROM public.ecr.aws/lambda/nodejs:20

Copy the function code to the LAMBDA_TASK_ROOT directory
This environment variable is provided by the lambda base image
COPY hello.js ${LAMBDA_TASK_ROOT}

Set the CMD to the function handler
CMD ["hello.handler"]
```

8. Buka `lib/hello.js` file dan tempel konten berikut ke dalamnya.

```
// define the handler
exports.handler = async (event) => {
 // simply return a friendly success response
 const response = {
 statusCode: 200,
 body: JSON.stringify('Hello, World!'),
 };
 return response;
};
```

9. Gunakan AWS CDK CLI untuk mensintesis proyek dan menyebarkan sumber daya. Anda harus bootstrap akun Anda.

```
npx cdk synth
npx cdk deploy --require-approval never
```

10. Panggil fungsi Lambda untuk mengonfirmasi dan memverifikasinya.

```
aws lambda invoke --function-name DockerTutorialFunction out.json
jq . out.json
```

Anda sekarang telah berhasil menerapkan fungsi Lambda berbasis kontainer Docker menggunakan AWS CDK. Untuk informasi selengkapnya tentang AWS CDK, lihat [Panduan Pengembangan AWS CDK v2](#). Jika Anda mengalami kesalahan atau mengalami masalah saat mencoba menyelesaikan tutorial ini, lihat bagian [Pemecahan Masalah](#) dari panduan ini untuk mendapatkan bantuan.

## Bersihkan

Anda sekarang telah berhasil menerapkan fungsi Lambda berbasis kontainer Docker menggunakan AWS CDK. Di dalam AWS CDK proyek, jalankan perintah berikut untuk menghapus sumber daya terkait. Anda akan diminta untuk mengkonfirmasi penghapusan.

- ```
npx cdk destroy DockerTutorialStack
```
- Untuk menghapus file dan sumber daya yang Anda buat dalam tutorial ini dari AWS CloudShell lingkungan Anda, jalankan perintah berikut.

```
cd ~  
rm -rf ~/docker-cli-tutorial
```

Bekerja dengan AWS CloudShell

Bagian ini menjelaskan cara berinteraksi dengan AWS CloudShell dan melakukan tindakan spesifik dengan aplikasi yang didukung.

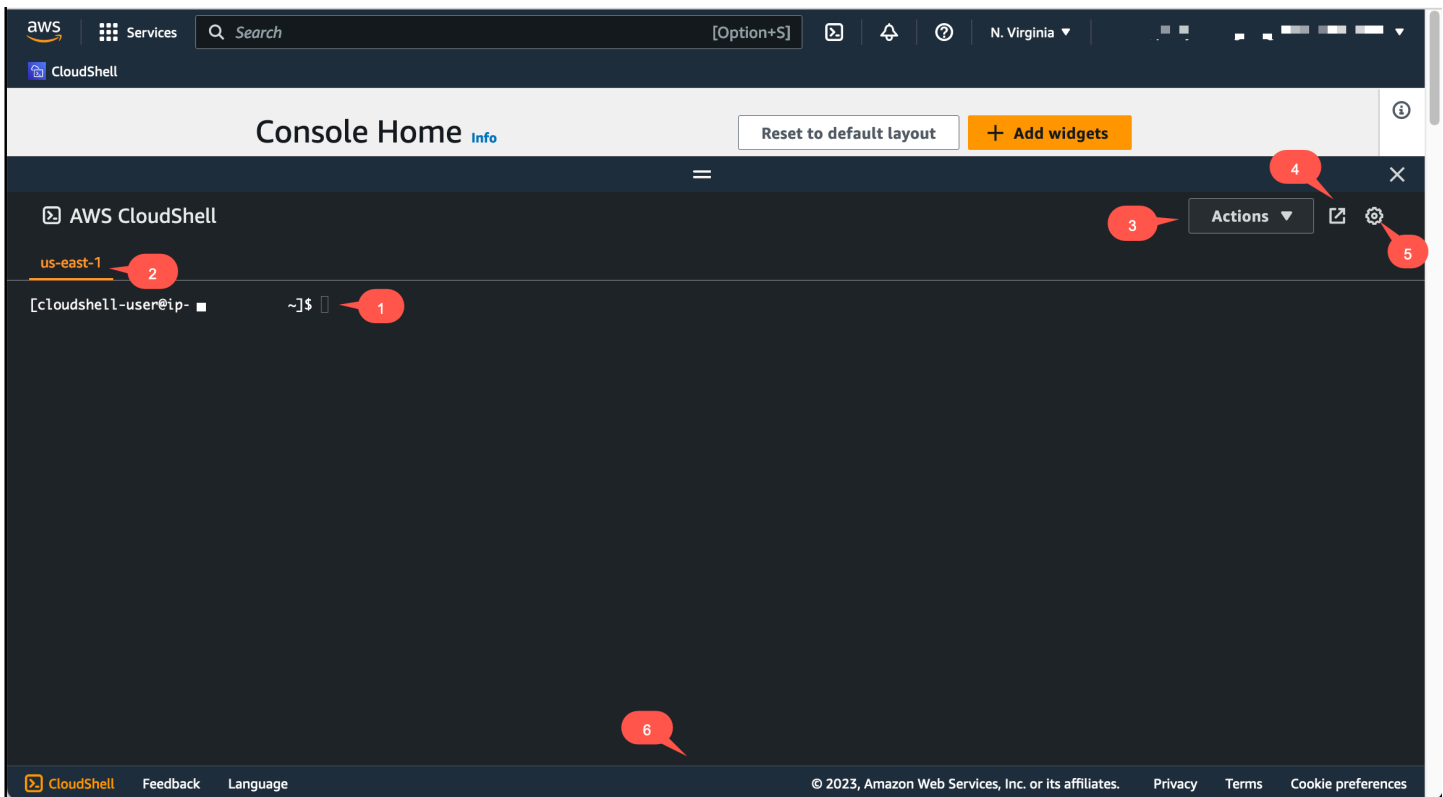
Topik

- [Menavigasi antarmuka AWS CloudShell](#)
- [Bekerja di Wilayah AWS](#)
- [Bekerja dengan file dan penyimpanan](#)
- [Bekerja dengan Docker](#)


Menavigasi antarmuka AWS CloudShell

Anda dapat menavigasi fitur CloudShell antarmuka dari AWS Management Console dan Console Toolbar.

Tangkapan layar berikut menunjukkan beberapa fitur AWS CloudShell antarmuka utama.




1. AWS CloudShell antarmuka baris perintah yang Anda gunakan untuk menjalankan perintah dengan menggunakan [shell pilihan Anda](#). Jenis shell saat ini ditunjukkan oleh command prompt.
2. Tab terminal, yang menggunakan Wilayah AWS where saat AWS CloudShell ini berjalan.
3. Menu Tindakan, yang menyediakan opsi untuk [mengubah tata letak layar](#), [mengunduh](#) dan [mengunggah](#) file, [memulai ulang AWS CloudShell](#), dan [menghapus direktori home Anda](#). AWS CloudShell

 Note

Opsi Unduh tidak tersedia saat Anda meluncurkan CloudShell di fileConsole Toolbar.

4. Buka di tab browser baru, yang menyediakan opsi untuk mengakses CloudShell sesi Anda dalam layar penuh.
5. Opsi Preferensi, yang dapat Anda gunakan untuk [menyesuaikan pengalaman shell Anda](#).
6. Bilah bawah, yang menyediakan opsi berikut untuk:
 - Luncurkan CloudShell dari CloudShell ikon.
 - Berikan umpan balik dari ikon Umpan Balik. Pilih jenis umpan balik yang ingin Anda kirimkan, tambahkan komentar Anda, lalu pilih Kirim.
 - Untuk mengirimkan umpan balik CloudShell, pilih salah satu opsi berikut:
 - Dari konsol, luncurkan CloudShell, dan pilih Umpan Balik. Tambahkan komentar Anda, lalu pilih Kirim.
 - Pilih CloudShell pada Console Toolbar, di kiri bawah konsol, dan kemudian pilih Buka di ikon tab browser baru, Umpan balik. Tambahkan komentar Anda, lalu pilih Kirim.

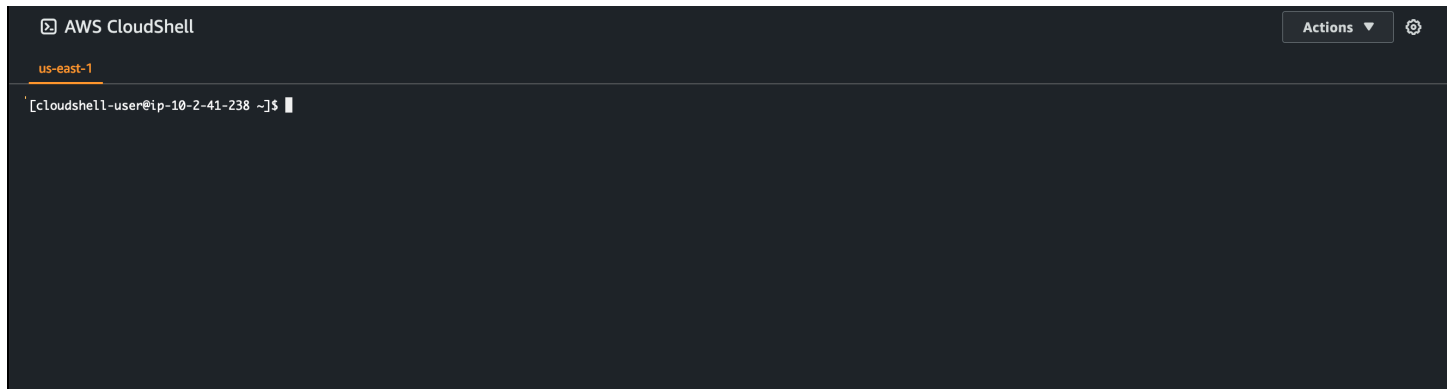
 Note

Opsi Umpan Balik tidak tersedia saat Anda meluncurkan CloudShell di Console Toolbar.

- Pelajari tentang kebijakan privasi dan ketentuan penggunaan kami, dan sesuaikan preferensi cookie.

Bekerja di Wilayah AWS

Arus Wilayah AWS yang Anda jalankan ditampilkan di atas antarmuka baris perintah.



```
AWS CloudShell
us-east-1
[cloudshell-user@ip-10-2-41-238 ~]$
```

Anda dapat memilih Wilayah AWS untuk bekerja dengan memilih Wilayah tertentu menggunakan pemilih Wilayah. Setelah Anda mengubah Regions, antarmuka akan diperbarui saat sesi shell Anda terhubung ke lingkungan komputasi berbeda yang berjalan di Wilayah yang dipilih.

Important

Anda dapat menggunakan hingga 1 GB penyimpanan persisten di masing-masing Wilayah AWS. Penyimpanan persisten disimpan di direktori home Anda (\$HOME). Ini berarti bahwa setiap file pribadi, direktori, program, atau skrip yang disimpan di direktori home Anda semuanya terletak dalam satu. Wilayah AWS Selain itu, mereka berbeda dari yang terletak di direktori home dan menyimpan Wilayah yang berbeda.

Retensi jangka panjang file dalam penyimpanan persisten juga dikelola berdasarkan Perwilayah. Untuk informasi selengkapnya, lihat [Penyimpanan tetap](#).

Menentukan default Wilayah AWS Anda untuk AWS CLI

Anda dapat menggunakan [variabel lingkungan](#) untuk menentukan opsi konfigurasi dan kredensial yang diperlukan untuk mengakses Layanan AWS menggunakan. AWS CLI Variabel lingkungan yang menentukan default Wilayah AWS untuk sesi shell Anda diatur baik ketika Anda memulai AWS CloudShell dari Wilayah tertentu di AWS Management Console atau ketika Anda memilih opsi di pemilih Region.

[Variabel lingkungan lebih diutamakan daripada file AWS CLI kredensial yang diperbarui oleh file.](#)

`aws configure` Jadi, Anda tidak dapat menjalankan `aws configure` perintah untuk mengubah Wilayah yang ditentukan oleh variabel lingkungan. Sebagai gantinya, untuk mengubah Region default untuk AWS CLI perintah, tetapkan nilai ke variabel `AWS_REGION` lingkungan. Dalam contoh berikut, ganti `us-east-1` dengan Wilayah tempat Anda berada.

Bash or Zsh

```
$ export AWS_REGION=us-east-1
```

Menyetel variabel lingkungan mengubah nilai yang digunakan hingga di akhir sesi shell Anda atau saat Anda menyetel variabel ke nilai yang berbeda. Anda dapat mengatur variabel dalam skrip startup shell Anda untuk membuat variabel persisten di seluruh sesi masa depan.

PowerShell

```
PS C:\> $Env:AWS_REGION="us-east-1"
```

Jika Anda menetapkan variabel lingkungan pada PowerShell prompt, variabel lingkungan menyimpan nilai hanya untuk durasi sesi saat ini. Atau, Anda dapat mengatur variabel untuk semua PowerShell sesi future dengan menambahkan variabel ke PowerShell profil Anda. Untuk informasi selengkapnya tentang menyimpan variabel lingkungan, lihat [PowerShell dokumentasi](#).

Untuk mengonfirmasi bahwa Anda telah mengubah Region default, jalankan `aws configure list` perintah untuk menampilkan data AWS CLI konfigurasi saat ini.

Note

Untuk AWS CLI perintah tertentu, Anda dapat mengganti Region default menggunakan opsi `--region` baris perintah. Untuk informasi selengkapnya, lihat [Opsis baris perintah](#) di Panduan AWS Command Line Interface Pengguna.

Bekerja dengan file dan penyimpanan

Menggunakan AWS CloudShell antarmuka, Anda dapat mengunggah file ke dan mengunduh file dari lingkungan shell. Untuk informasi selengkapnya tentang mengunduh dan mengunggah file, lihat [Memulai AWS CloudShell](#).

Untuk memastikan file apa pun yang Anda tambahkan tersedia setelah sesi Anda berakhir, Anda harus mengetahui perbedaan antara penyimpanan persisten dan sementara.

- Penyimpanan persisten: Anda memiliki 1 GB penyimpanan persisten untuk masing-masing Wilayah AWS. Penyimpanan persisten ada di direktori home Anda.

- Penyimpanan sementara: Penyimpanan sementara didaur ulang pada akhir sesi. Penyimpanan sementara ada di direktori yang berada di luar direktori home Anda.

Important

Pastikan untuk meninggalkan file yang ingin Anda simpan dan gunakan untuk sesi shell future di direktori home Anda. Misalnya, Anda memindahkan file dari direktori home Anda dengan menjalankan mv perintah. Kemudian, file itu didaur ulang ketika sesi shell saat ini berakhir.

Bekerja dengan Docker

AWS CloudShell sepenuhnya mendukung Docker tanpa instalasi atau konfigurasi. Anda dapat menentukan, membangun, dan menjalankan kontainer Docker di dalamnya AWS CloudShell. Anda dapat menerapkan sumber daya berbasis Docker, seperti fungsi Lambda berdasarkan kontainer Docker, melalui AWS CDK Toolkit serta membangun kontainer Docker dan mendorongnya ke repositori Amazon ECR melalui CLI Docker. Untuk langkah-langkah mendetail tentang cara menjalankan kedua penerapan ini, lihat tutorial berikut:

- [Tutorial: Menyebarkan fungsi Lambda menggunakan AWS CDK](#)
- [Tutorial: Membangun wadah Docker di dalam AWS CloudShell dan mendorongnya ke repositori Amazon ECR](#)

Ada batasan dan batasan tertentu dengan menggunakan Docker dengan AWS CloudShell:

- Docker memiliki ruang terbatas di suatu lingkungan. Jika Anda memiliki gambar individual yang besar, atau terlalu banyak gambar Docker yang sudah ada sebelumnya, ini dapat menyebabkan masalah yang mungkin mencegah Anda menarik, membangun, atau menjalankan gambar tambahan. Untuk informasi selengkapnya tentang Docker, lihat panduan [Dokumentasi Docker](#).
- Docker hanya didukung di Wilayah tertentu. Untuk informasi tentang Wilayah mana yang didukung dengan Docker, lihat Wilayah [Docker](#).
- Jika Anda mengalami masalah saat menggunakan Docker AWS CloudShell, lihat bagian [Pemecahan Masalah](#) di panduan ini untuk informasi tentang cara mengatasi masalah ini secara potensial.

Bekerja dengan fitur aksesibilitas untuk AWS CloudShell

Topik ini menjelaskan cara menggunakan fitur aksesibilitas untuk CloudShell. Anda dapat menggunakan keyboard untuk menavigasi melalui elemen yang dapat difokuskan pada halaman. Anda juga dapat menyesuaikan tampilan CloudShell, termasuk ukuran font dan tema antarmuka.

Navigasi keyboard di CloudShell

Untuk menavigasi melalui elemen yang dapat difokuskan pada halaman, tekan Tab.

CloudShell fitur aksesibilitas terminal

Anda dapat menggunakan Tab kunci dalam mode berikut:

- Modus terminal (Default)— Dalam mode ini, terminal menangkap Tab enter kunci. Setelah fokus pada terminal, tekan Tab untuk mengakses hanya fungsionalitas terminal.
- Modus navigasi— Dalam mode ini, terminal tidak menangkap Tab enter kunci. Per Tab untuk menavigasi melalui elemen yang dapat difokuskan pada halaman.

Untuk beralih antara mode terminal dan mode navigasi, tekan Ctrl+M. Setelah Anda beralih kembali, Tab: navigasi muncul di header, dan Anda dapat menggunakan Tab kunci untuk menavigasi melalui halaman.

Untuk kembali ke mode terminal, tekan Ctrl+M. Atau, pilih X di sebelah Tab: navigasi.

Note

Saat ini, CloudShell fitur aksesibilitas terminal tidak tersedia di perangkat seluler.

Memilih ukuran font dan tema antarmuka di CloudShell

Anda dapat menyesuaikan tampilan CloudShell untuk mengakomodasi preferensi visual Anda.

- Ukuran font— Pilih dari Terkecil, Kecil, Sedang, Besar, dan Terbesar ukuran font di terminal. Untuk informasi selengkapnya tentang mengubah ukuran font, lihat [the section called “Mengubah ukuran font”](#).

- Tema— Pilih antara Cahaya dan Gelap tema antarmuka. Untuk informasi selengkapnya tentang mengubah tema antarmuka, lihat [the section called “Mengubah tema antarmuka”](#).

Bekerja dengan AWS layanan di AWS CloudShell

Manfaat utama AWS CloudShell adalah Anda dapat menggunakannya untuk mengelola AWS layanan Anda dari antarmuka baris perintah. Ini berarti bahwa Anda tidak perlu mengunduh dan menginstal alat atau mengkonfigurasi kredensial Anda sebelumnya. Saat Anda memulai AWS CloudShell, lingkungan komputasi dibuat yang memiliki alat baris perintah AWS berikut yang sudah diinstal:

- [AWS CLI](#)
- [AWS Elastic Beanstalk CLI](#)
- [Amazon ECS CLI](#)
- [AWS SAM](#)

Dan karena Anda sudah masuk AWS, tidak ada persyaratan untuk mengonfigurasi kredensial Anda secara lokal sebelum menggunakan layanan. Kredensial yang Anda gunakan untuk masuk ke AWS Management Console diteruskan ke AWS CloudShell.

Jika Anda ingin mengubah AWS Wilayah default yang digunakan AWS CLI, Anda dapat mengubah nilai yang ditetapkan ke variabel `AWS_REGION` lingkungan. (Untuk informasi selengkapnya, lihat [Menentukan default Wilayah AWS Anda untuk AWS CLI](#).)

Sisa topik ini menunjukkan bagaimana Anda dapat mulai menggunakan AWS CloudShell untuk berinteraksi dengan AWS layanan yang dipilih dari baris perintah.

AWS CLI contoh baris perintah untuk AWS layanan yang dipilih

Contoh berikut hanya mewakili beberapa dari banyak AWS layanan yang dapat Anda gunakan menggunakan perintah yang tersedia dari AWS CLI Versi 2. Untuk daftar lengkap, lihat [Referensi Perintah AWS CLI](#).

- [DynamoDB](#)
- [AWS Cloud9](#)
- [Amazon EC2](#)
- [Gletser S3](#)

DynamoDB

DynamoDB adalah layanan basis data NoSQL terkelola penuh yang memberikan performa yang cepat dan dapat diprediksi dengan skalabilitas sempurna. Implementasi layanan ini dari mode NoSQL mendukung kunci-nilai dan struktur data dokumen.

`create-table` Perintah berikut membuat tabel bergaya NoSQL yang diberi nama `MusicCollection` di AWS akun Anda.

```
aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=Artist,AttributeType=S \  
  AttributeName=SongTitle,AttributeType=S \  
  --key-schema AttributeName=Artist,KeyType=HASH \  
  AttributeName=SongTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \  
  --tags Key=Owner,Value=blueTeam
```

Untuk informasi selengkapnya, lihat [Menggunakan DynamoDB dengan AWS CLI](#) di Panduan AWS Command Line Interface Pengguna.

AWS Cloud9

AWS Cloud9 adalah integrated development environment (IDE) berbasis cloud yang dapat Anda gunakan untuk menulis, menjalankan, dan men-debug kode Anda di jendela peramban. Lingkungan memiliki editor kode, debugger, dan terminal.

`create-environment-ec2` Perintah berikut menciptakan lingkungan pengembangan AWS Cloud9 EC2 dengan pengaturan yang ditentukan. Mesin tersebut meluncurkan instans Amazon EC2, dan kemudian terhubung dari instans ke lingkungan.

```
aws cloud9 create-environment-ec2 --name my-demo-env --description "My demonstration development environment." --instance-type t2.micro --subnet-id subnet-1fab8aEX --automatic-stop-time-minutes 60 --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

Untuk informasi lebih lanjut, lihat [referensi AWS Cloud9 baris perintah](#).

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang aman dan ukurannya dapat diubah-ubah di cloud. Mesin tersebut dirancang untuk mempermudah cloud computing skala web dan lebih mudah diakses.

`run-instances` Perintah berikut meluncurkan instance `t2.micro` di subnet VPC yang ditentukan:

```
aws ec2 run-instances --image-id ami-xxxxxxx --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

Untuk informasi selengkapnya, lihat [Menggunakan Amazon EC2 dengan AWS CLI](#) di Panduan AWS Command Line Interface Pengguna.

S3 Glacier

S3 Glacier dan S3 Glacier Deep Archive adalah kelas penyimpanan cloud Amazon S3 yang aman, tahan lama, dan berbiaya sangat rendah untuk pengarsipan data dan cadangan jangka panjang.

`create-vault` Perintah berikut membuat vault — kontainer untuk menyimpan arsip:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Untuk informasi selengkapnya, lihat [Menggunakan Amazon S3 Glacier dengan AWS CLI](#) di Panduan AWS Command Line Interface Pengguna.

AWS Elastic Beanstalk

AWS Elastic Beanstalk CLI menyediakan antarmuka baris perintah yang dibuat untuk menyederhanakan pembuatan, pembaruan, dan pemantauan lingkungan dari repositori lokal. Dalam konteks ini, lingkungan mengacu pada koleksi AWS sumber daya yang menjalankan versi aplikasi.

`create` Perintah berikut membuat lingkungan baru di Amazon Virtual Private Cloud (VPC) kustom.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --vpc.securitygroup sg-70cff265
```

Untuk informasi selengkapnya, lihat [referensi perintah EB CLI](#) di Panduan AWS Elastic Beanstalk Pengembang.

Amazon ECS

Antarmuka baris perintah (CLI) Amazon ECS) menyediakan beberapa perintah tingkat tinggi. Mesin tersebut dirancang untuk menyederhanakan proses pembuatan, pembaruan, serta pemantauan kluster dan tugas dari lingkungan pengembangan lokal. (Kluster Amazon ECS adalah pengelompokan logis dari tugas atau layanan.)

Perintah berikut mengonfigurasi CLI Amazon ECS untuk membuat konfigurasi kluster bernama `ecs-cli-demo`. Konfigurasi kluster ini digunakan `FARGATE` sebagai tipe peluncuran default untuk `ecs-cli-demo` kluster di `us-east-1` region.

```
ecs-cli configure --region us-east-1 --cluster ecs-cli-demo --default-launch-type
FARGATE --config-name ecs-cli-demo
```

Untuk informasi lebih lanjut, lihat [Referensi Baris Perintah Amazon ECS dalam Panduan Developer Amazon ECS Command Line](#) dalam Panduan Developer Amazon ECS Container Service.

CLI AWS SAM

AWS SAM CLI adalah alat baris perintah yang beroperasi pada AWS Serverless Application Model templat dan kode aplikasi. Anda dapat melakukan beberapa tugas menggunakannya. Ini termasuk memanggil fungsi Lambda lokal, membuat paket deployment pada aplikasi deployment pada aplikasi Lambda, dan men-deploy aplikasi Lambda, dan men-deploy aplikasi Lambda lokal, membuat paket deployment pada aplikasi deployment, dan men-deploy aplikasi Lambda, dan deployment aplikasi Lambda lokal, membuat AWS paket deployment

Perintah berikut menginisialisasi proyek SAM baru dengan parameter yang diperlukan dilewatkan sebagai parameter:

```
sam init --runtime python3.7 --dependency-manager pip --app-template hello-world --name
sam-app
```

Untuk informasi selengkapnya, lihat [referensi perintah AWS SAM CLI](#) di Panduan AWS Serverless Application Model Pengembang.

Menyesuaikan AndaAWS CloudShellpengalaman

Anda dapat menyesuaikan aspek-aspek berikutAWS CloudShellpengalaman:

- [Tata letak tab](#): Pisahkan antarmuka baris perintah menjadi beberapa kolom dan baris.
- [Ukuran font](#): Sesuaikan ukuran teks baris perintah.
- [Tema warna](#): Beralih antara tema terang dan gelap.
- [Pasta Aman](#): Aktifkan atau nonaktifkan fitur yang mengharuskan Anda memverifikasi teks multiline sebelum ditempelkan.
- [Tmux ke pemulihan sesi](#): Menggunakan tmux mengembalikan sesi Anda hingga sesi menjadi tidak aktif.

Anda juga dapat memperluas lingkungan shell Anda dengan[menginstal perangkat lunak Anda sendiri](#)dan[memodifikasi skrip shell start-up](#).

Memisahkan tampilan baris perintah menjadi beberapa tab

Jalankan beberapa perintah dengan memisahkan antarmuka baris perintah Anda menjadi beberapa panel.

Note

Setelah membuka beberapa tab, Anda dapat memilih salah satu yang ingin Anda kerjakan dengan mengklik di mana saja di panel pilihan Anda. Anda dapat menutup tab dengan memilih simbol, yang berada di sebelah nama Wilayah.

- PilihTindakan salah satu opsi berikut dariTata letak tab:
 - Tab baru: Tambahkan tab baru yang berada di sebelah tab yang sedang aktif.
 - Pisahkan menjadi beberapa baris: Tambahkan tab baru dalam satu baris yang berada di bawah tab yang sedang aktif.
 - Dibagi menjadi kolom: Tambahkan tab baru di kolom yang berada di sebelah yang sedang aktif.

Jika tidak ada cukup ruang untuk menampilkan setiap tab sepenuhnya, gulir untuk melihat seluruh tab. Anda juga dapat memilih bilah terpisah yang memisahkan panel dan menyeretnya dengan menggunakan penunjuk untuk menambah atau mengurangi ukuran panel.

Mengubah ukuran font

Menambah atau mengurangi ukuran teks yang ditampilkan di antarmuka baris perintah.

1. Untuk mengubah AWS CloudShell pengaturan terminal, buka Pengaturan, Preferensi.
2. Pilih ukuran teks. Pilihan Anda adalah Terkecil, Kecil, Sedang, Besar, dan Terbesar.

Mengubah tema antarmuka

Beralih antara tema terang dan gelap untuk antarmuka baris perintah.

1. Untuk mengubah AWS CloudShell tema, pergi ke Pengaturan, Preferensi.
2. Pilih Cahaya atau Gelap.

Menggunakan Safe Paste untuk teks multiline

Safe Paste adalah fitur keamanan yang meminta Anda untuk memverifikasi bahwa teks multiline yang akan Anda tempel ke dalam shell tidak mengandung skrip berbahaya. Teks yang disalin dari situs pihak ketiga dapat berisi kode tersembunyi yang memicu perilaku tak terduga di lingkungan shell Anda.

Dialog Safe Paste menampilkan teks lengkap yang Anda salin ke clipboard Anda. Jika Anda puas bahwa tidak ada risiko keamanan, pilih Tempel.

Warning: Pasting multiline text into AWS CloudShell

Text that's copied from external sources can contain malicious scripts. Verify the text below before pasting.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
total=x+y+z
print("The total is",total)
```

Always ask before pasting multiline code

Cancel

Paste

Kami menyarankan Anda mengaktifkan Safe Paste untuk menangkap potensi risiko keamanan dalam skrip. Anda dapat mengaktifkan atau menonaktifkan fitur ini dengan memilih Preferensi, Aktifkan Tempel Aman dan Nonaktifkan Safe Paste.

Menggunakan tmux ke pemulihan sesi

AWS CloudShell menggunakan tmux untuk memulihkan sesi di satu atau beberapa tab browser. Jika Anda me-refresh tab browser, itu melanjutkan sesi Anda sampai sesi menjadi tidak aktif. Untuk informasi lebih lanjut, lihat [Pemulihan sesi](#).

Keamanan untuk AWS CloudShell

Keamanan cloud di Amazon Web Services (AWS) merupakan prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan. Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model Tanggung Jawab Bersama](#) menggambarkan ini sebagai Keamanan dari Cloud dan Keamanan dalam Cloud.

Security of the Cloud - AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan di AWS Cloud dan memberi Anda layanan yang dapat Anda gunakan dengan aman. Tanggung jawab keamanan kami adalah prioritas tertinggi di AWS, dan efektivitas keamanan kami secara teratur diuji dan diverifikasi oleh auditor pihak ketiga sebagai bagian dari [Program AWS Kepatuhan](#).

Keamanan di Cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan, dan faktor-faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, serta undang-undang dan peraturan yang berlaku.

AWS CloudShell mengikuti [model tanggung jawab bersama](#) melalui AWS layanan khusus yang didukungnya. Untuk informasi keamanan AWS layanan, lihat [halaman dokumentasi keamanan AWS layanan](#) dan [AWS layanan yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

Topik berikut menunjukkan cara mengonfigurasi AWS CloudShell untuk memenuhi tujuan keamanan dan kepatuhan Anda.

Topik

- [Perlindungan data di AWS CloudShell](#)
- [Identity and Access Management untuk AWS CloudShell](#)
- [Penebangan dan pemantauan di AWS CloudShell](#)
- [Validasi kepatuhan untuk AWS CloudShell](#)
- [Ketahanan di AWS CloudShell](#)
- [Keamanan infrastruktur di AWS CloudShell](#)
- [Analisis konfigurasi dan kerentanan di AWS CloudShell](#)
- [Praktik terbaik keamanan untuk AWS CloudShell](#)

- [AWS CloudShell FAQ Keamanan](#)

Perlindungan data di AWS CloudShell

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di AWS CloudShell. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan AWS CloudShell atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi data

Enkripsi data mengacu pada melindungi data saat diam (saat disimpan di AWS CloudShell) dan saat transit (saat bergerak antara AWS CloudShell dan titik akhir layanan).

Enkripsi saat istirahat menggunakan AWS KMS

Enkripsi saat istirahat didefinisikan sebagai perlindungan data dari akses tidak sah dengan mengenkripsi data saat disimpan. Saat menggunakan AWS CloudShell, Anda memiliki penyimpanan persisten 1 GB per AWS Wilayah tanpa biaya. Penyimpanan persisten terletak di direktori home Anda (\$HOME) dan bersifat pribadi untuk Anda. Tidak seperti sumber daya lingkungan sementara yang didaur ulang setelah setiap sesi shell berakhir, data di direktori home Anda tetap ada.

Enkripsi data yang disimpan AWS CloudShell diimplementasikan menggunakan kunci kriptografi yang disediakan oleh AWS Key Management Service (AWS KMS). Ini adalah AWS layanan terkelola untuk membuat dan mengendalikan kunci master pelanggan (CMK) —kunci enkripsi yang digunakan untuk mengenkripsi data pelanggan yang disimpan di lingkungan. AWS CloudShell menghasilkan dan mengelola kunci kriptografi untuk mengenkripsi data atas nama pelanggan.

Enkripsi dalam bergerak

Enkripsi dalam transit didefinisikan sebagai perlindungan data dari intersepsi saat data ditransfer antar-endpoint komunikasi.

Secara default, semua komunikasi data antara komputer browser web klien dan berbasis cloud AWS CloudShell dienkripsi dengan mengirimkan semuanya melalui koneksi HTTPS/TLS.

Anda tidak perlu melakukan apa pun untuk mengaktifkan penggunaan HTTPS/TLS untuk komunikasi.

Identity and Access Management untuk AWS CloudShell

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang

dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. CloudShell IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana AWS CloudShell bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk AWS CloudShell](#)
- [Memecahkan masalah CloudShell identitas dan akses AWS](#)
- [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. CloudShell

Pengguna layanan — Jika Anda menggunakan CloudShell layanan untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensial dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak CloudShell fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di CloudShell, lihat [Memecahkan masalah CloudShell identitas dan akses AWS](#).

Administrator layanan — Jika Anda bertanggung jawab atas CloudShell sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke CloudShell. Tugas Anda adalah menentukan CloudShell fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM CloudShell, lihat [Bagaimana AWS CloudShell bekerja dengan IAM](#).

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke CloudShell. Untuk melihat contoh kebijakan CloudShell berbasis identitas yang dapat Anda gunakan di IAM, lihat [Contoh kebijakan berbasis identitas untuk AWS CloudShell](#)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) di AWS](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar tugas lengkap

yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensial sementara.

Untuk pengelolaan akses terpusat, sebaiknya Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat [Apa yang dimaksud Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial temporer, dan bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, sebaiknya rotasikan kunci akses. Untuk informasi selengkapnya, lihat [Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat diambil oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk

mempelajari selengkapnya, silakan lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi diautentikasi, identitas tersebut dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi sekumpulan izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengaitkan izin yang ditetapkan ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat [Rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Contoh, ketika Anda melakukan panggilan dalam layanan, umumnya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Meneruskan sesi akses](#).
- Peran IAM – Peran layanan adalah [peran IAM](#) yang diambil layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat mengambil peran untuk melakukan sebuah tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan

diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Ikhtisar kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat [Memilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya.

Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

Tipe kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- **Batasan izin** – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang Principal tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- **Kebijakan kontrol layanan (SCP)** — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke sebagian atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .

- **Kebijakan sesi** – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit di salah satu kebijakan ini akan membatalkan izin tersebut. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Jika beberapa jenis kebijakan diberlakukan untuk satu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana AWS CloudShell bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses CloudShell, pelajari fitur IAM yang tersedia untuk digunakan. CloudShell

Fitur IAM yang dapat Anda gunakan dengan AWS CloudShell

Fitur IAM	CloudShell dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACL	Tidak
ABAC (tanda dalam kebijakan)	Tidak
Kredensial sementara	Ya

Fitur IAM	CloudShell dukungan
Sesi akses teruskan (FAS)	Tidak
Peran layanan	Tidak
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara CloudShell dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk CloudShell

Mendukung kebijakan berbasis identitas	Ya
--	----

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta ketentuan terkait jenis tindakan yang diizinkan atau ditolak. Anda tidak dapat menentukan pengguna utama dalam kebijakan berbasis identitas karena kebijakan ini berlaku untuk pengguna atau peran yang dilampiri kebijakan. Untuk mempelajari semua elemen yang dapat digunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk CloudShell

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS CloudShell](#)

Kebijakan berbasis sumber daya dalam CloudShell

Mendukung kebijakan berbasis sumber daya	Tidak
--	-------

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Izin diberikan dengan melampirkan kebijakan berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, kebijakan berbasis identitas lainnya tidak diperlukan. Untuk informasi selengkapnya, lihat [Perbedaan peran IAM dengan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Tindakan kebijakan untuk CloudShell

Mendukung tindakan kebijakan

Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Untuk melihat daftar CloudShell tindakan, lihat [Tindakan yang ditentukan oleh AWS CloudShell](#) di Referensi Otorisasi Layanan. Beberapa tindakan mungkin memiliki lebih dari satu API.

Tindakan kebijakan CloudShell menggunakan awalan berikut sebelum tindakan:

```
cloudshell
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut dengan koma.

```
"Action": [  
  "cloudshell:action1",  
  "cloudshell:action2"  
]
```

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS CloudShell](#)

Sumber daya kebijakan untuk CloudShell

Mendukung sumber daya kebijakan	Ya
---------------------------------	----

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk mengindikasikan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*" 
```

Untuk melihat daftar jenis CloudShell sumber daya dan ARNnya, lihat Sumber [daya yang ditentukan oleh AWS CloudShell](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh AWS](#). CloudShell

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS CloudShell](#)

Kunci kondisi kebijakan untuk CloudShell

Mendukung kunci kondisi kebijakan spesifik layanan	Ya
--	----

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen Condition (atau blok Condition) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen Condition bersifat opsional. Anda dapat membuat ekspresi kondisional yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen Condition dalam satu pernyataan, atau beberapa kunci dalam satu elemen Condition, AWS akan mengevaluasinya dengan menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat [Elemen kebijakan IAM: variabel dan tanda](#) di Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci CloudShell kondisi, lihat [Kunci kondisi untuk AWS CloudShell](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh AWS CloudShell](#).

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS CloudShell](#)

ACL di CloudShell

Mendukung ACL

Tidak

Daftar kontrol akses (ACL) mengontrol pengguna utama (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan CloudShell

Mendukung ABAC (tanda dalam kebijakan)

Tidak

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Pemberian tanda ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi-operasi ketika tanda milik pengguna utama cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna dalam situasi di mana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tanda di [elemen syarat](#) dari sebuah kebijakan dengan menggunakan kunci-kunci persyaratan `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi hanya untuk beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat [Apa itu ABAC?](#) di Panduan Pengguna IAM. Untuk melihat tutorial terkait langkah-langkah penyiapan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) di Panduan Pengguna IAM.

Menggunakan kredensial sementara dengan CloudShell

Mendukung kredensial sementara

Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensial sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensial sementara, lihat [Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Anda menggunakan kredensial sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensial sementara. Anda juga akan membuat kredensial sementara secara otomatis saat masuk ke konsol sebagai pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang cara beralih peran, lihat [Beralih peran \(konsol\)](#) di Panduan Pengguna IAM.

Anda dapat membuat kredensial sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensial sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#).

Ketika Anda beralih peran, Anda akan menggunakan lingkungan yang berbeda. Anda tidak dapat beralih peran dalam AWS CloudShell lingkungan yang sama.

Teruskan sesi akses untuk CloudShell

Mendukung sesi akses maju (FAS)

Tidak

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk

menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Meneruskan sesi akses](#).

Peran layanan untuk CloudShell

Mendukung peran layanan

Tidak

Peran layanan adalah sebuah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Warning

Mengubah izin untuk peran layanan dapat merusak CloudShell fungsionalitas. Edit peran layanan hanya jika CloudShell memberikan panduan untuk melakukannya.

Peran terkait layanan untuk CloudShell

Mendukung peran terkait layanan

Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat mengambil peran untuk melakukan sebuah tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Contoh kebijakan berbasis identitas untuk AWS CloudShell

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi CloudShell sumber daya. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh CloudShell, termasuk format ARN untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk AWS CloudShell](#) di Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan konsol CloudShell](#)
- [Izinkan pengguna melihat izin mereka sendiri](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus CloudShell sumber daya di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [kebijakan yang dikelola AWS](#) atau [kebijakan yang dikelola AWS untuk fungsi pekerjaan](#) di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat [Kebijakan dan izin di IAM](#) di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan

akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.

- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [validasi kebijakan Analizer Akses IAM](#) di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi akses API yang dilindungi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) di Panduan Pengguna IAM.

Menggunakan konsol CloudShell

Untuk mengakses CloudShell konsol AWS, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang CloudShell sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebaliknya, izinkan akses hanya ke tindakan yang cocok dengan operasi API yang coba dilakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan CloudShell konsol, lampirkan juga kebijakan CloudShell *ConsoleAccess* atau *ReadOnly* AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan Pengguna IAM.

Izinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini

mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Memecahkan masalah CloudShell identitas dan akses AWS

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan CloudShell dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di CloudShell](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses CloudShell sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di CloudShell

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `aws:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `aws:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran CloudShell.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di CloudShell. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses CloudShell sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah CloudShell mendukung fitur-fitur ini, lihat [Bagaimana AWS CloudShell bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Memberikan akses kepada pengguna eksternal yang sah \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan antara peran IAM dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM

Dengan sumber daya manajemen akses yang dapat disediakan oleh AWS Identity and Access Management (IAM), administrator dapat memberikan izin kepada pengguna IAM. Dengan begitu, pengguna ini dapat mengakses AWS CloudShell dan menggunakan fitur lingkungan. Administrator juga dapat membuat kebijakan yang menentukan pada tingkat terperinci tindakan apa yang dapat dilakukan pengguna tersebut dengan lingkungan shell.

Cara tercepat bagi administrator untuk memberikan akses ke pengguna adalah melalui kebijakan AWS terkelola. [Kebijakan terkelola AWS](#) adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. Kebijakan AWS terkelola berikut ini AWS CloudShell dapat dilampirkan ke identitas IAM:

- **AWS CloudShellFullAccess**: Memberikan izin untuk menggunakan AWS CloudShell dengan akses penuh ke semua fitur.

AWS CloudShellFullAccessKebijakan ini menggunakan karakter wildcard (*) untuk memberikan identitas IAM (pengguna, peran, atau grup) akses penuh ke CloudShell dan fitur. Untuk informasi selengkapnya tentang kebijakan ini, lihat [AWS CloudShellFullAccess](#) di Panduan Pengguna Kebijakan AWS Terkelola.

Note

Identitas IAM dengan kebijakan AWS terkelola berikut juga dapat diluncurkan. CloudShell Namun, kebijakan ini memberikan izin ekstensif. Jadi, kami menyarankan Anda hanya memberikan kebijakan ini jika kebijakan tersebut penting untuk peran pekerjaan pengguna IAM.

- **[Administrator](#)**: Menyediakan pengguna IAM dengan akses penuh dan memungkinkan mereka untuk mendelegasikan izin ke setiap layanan dan sumber daya di AWS
- **[Pengguna daya pengembang](#)**: Memungkinkan pengguna IAM untuk melakukan tugas pengembangan aplikasi dan membuat serta mengonfigurasi sumber daya dan layanan yang mendukung pengembangan aplikasi yang AWS sadar.

Untuk informasi selengkapnya tentang melampirkan kebijakan terkelola, lihat [Menambahkan izin identitas IAM \(konsol\) di Panduan Pengguna IAM](#).

Mengelola tindakan yang diizinkan dalam AWS CloudShell menggunakan kebijakan khusus

Untuk mengelola tindakan yang dapat dilakukan pengguna IAM CloudShell, buat kebijakan kustom yang menggunakan kebijakan CloudShellPolicy terkelola sebagai templat. Atau, edit [kebijakan inline](#) yang disematkan dalam identitas IAM yang relevan (pengguna, grup, atau peran).

Misalnya, Anda dapat mengizinkan pengguna IAM untuk mengakses CloudShell, tetapi mencegah mereka meneruskan kredensial CloudShell lingkungan yang digunakan untuk masuk. AWS Management Console

Important

Untuk memulai AWS CloudShell dari AWS Management Console, pengguna IAM memerlukan izin untuk tindakan berikut:

- `CreateEnvironment`
- `CreateSession`
- `GetEnvironmentStatus`
- `StartEnvironment`

Jika salah satu tindakan ini tidak diizinkan secara eksplisit oleh kebijakan terlampir, kesalahan izin IAM akan ditampilkan saat Anda mencoba meluncurkan CloudShell

AWS CloudShell izin

Nama	Deskripsi izin yang diberikan	Diperlukan untuk meluncurkan CloudShell?
<code>cloudshell:CreateEnvironment</code>	Menciptakan CloudShell lingkungan, mengambil tata letak di awal	Ya

Nama	Deskripsi izin yang diberikan	Diperlukan untuk meluncurkan CloudShell?
	CloudShell sesi, dan menyimpan tata letak saat ini dari aplikasi web di backend. Izin ini hanya mengharapkan * sebagai nilai untuk Resource sebagaimana diuraikan dalam the section called “Contoh kebijakan IAM untuk CloudShell”	
cloudshell:CreateSession	Terhubung ke CloudShell lingkungan dari AWS Management Console.	Ya
cloudshell:GetEnvironmentStatus	Baca status CloudShell lingkungan.	Ya
cloudshell>DeleteEnvironment	Menghapus CloudShell lingkungan.	Tidak
cloudshell:GetFileDownloadUrls	Menghasilkan URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunduh file CloudShell melalui menggunakan CloudShell antarmuka web.	Tidak

Nama	Deskripsi izin yang diberikan	Diperlukan untuk meluncurkan CloudShell?
<code>cloudshell:GetFileUploadUrls</code>	Menghasilkan URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunggah file CloudShell melalui menggunakan CloudShell antarmuka web.	Tidak
<code>cloudshell:PutCredentials</code>	Meneruskan kredensial yang digunakan untuk masuk ke ke. AWS Management Console CloudShell	Tidak
<code>cloudshell:StartEnvironment</code>	Memulai CloudShell lingkungan yang diidentifikasi.	Ya
<code>cloudshell:StopEnvironment</code>	Menghentikan CloudShell lingkungan yang sedang berjalan.	Tidak

Contoh kebijakan IAM untuk CloudShell

Contoh berikut menunjukkan bagaimana kebijakan dapat dibuat untuk membatasi siapa yang dapat mengakses CloudShell. Contoh juga menunjukkan tindakan yang dapat dilakukan di lingkungan shell.

Kebijakan berikut ini memberlakukan penolakan lengkap atas akses CloudShell dan fitur-fiturnya.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DenyCloudShell",
    "Effect": "Deny",
    "Action": [
```

```

        "cloudshell:*"
    ],
    "Resource": "*"
  }]
}

```

Kebijakan berikut ini memungkinkan pengguna IAM untuk mengakses CloudShell tetapi memblokir mereka dari membuat URL yang telah ditandatangani sebelumnya untuk mengunggah dan mengunduh file. Pengguna masih dapat mentransfer file ke dan dari lingkungan, menggunakan klien seperti wget misalnya.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUsingCloudshell",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyUploadDownload",
      "Effect": "Deny",
      "Action": [
        "cloudshell:GetFileDownloadUrls",
        "cloudshell:GetFileUploadUrls"
      ],
      "Resource": "*"
    }
  ]
}

```

Kebijakan berikut memungkinkan pengguna IAM untuk mengakses CloudShell. Namun, kebijakan ini mencegah kredensial yang Anda gunakan untuk masuk ke AWS Management Console agar tidak diteruskan ke lingkungan. CloudShell Pengguna IAM dengan kebijakan ini perlu mengonfigurasi kredensialnya secara manual di dalamnya. CloudShell

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```



```

    "Sid": "AllowUsingCloudshell",
    "Effect": "Allow",
    "Action": [
        "cloudshell:*"
    ],
    "Resource": "*"
},
{
    "Sid": "DenyCredentialForwarding",
    "Effect": "Deny",
    "Action": [
        "cloudshell:PutCredentials"
    ],
    "Resource": "*"
}]
}

```

Kebijakan berikut memungkinkan pengguna IAM untuk membuat AWS CloudShell lingkungan.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CloudShellUser",
    "Effect": "Allow",
    "Action": [
      "cloudshell:CreateEnvironment",
      "cloudshell:CreateSession",
      "cloudshell:GetEnvironmentStatus",
      "cloudshell:StartEnvironment"
    ],
    "Resource": "*"
  }]
}

```

Izin untuk mengakses Layanan AWS

CloudShell menggunakan kredensial IAM yang Anda gunakan untuk masuk ke. AWS Management Console

Note

Untuk menggunakan kredensi IAM yang Anda gunakan untuk masuk AWS Management Console, Anda harus memiliki izin. `cloudshell:PutCredentials`

Fitur pra-otentikasi ini CloudShell membuatnya nyaman untuk digunakan. AWS CLI Namun, pengguna IAM masih memerlukan izin eksplisit untuk Layanan AWS yang dipanggil dari baris perintah.

Misalnya, pengguna IAM diharuskan membuat bucket Amazon S3 dan mengunggah file sebagai objek kepada mereka. Anda dapat membuat kebijakan yang secara eksplisit mengizinkan tindakan tersebut. Konsol IAM menyediakan [editor visual](#) interaktif yang memandu melalui proses membangun dokumen kebijakan berformat JSON. Setelah kebijakan dibuat, Anda dapat melampirkannya ke identitas IAM yang relevan (pengguna, grup, atau peran).

Untuk informasi selengkapnya tentang melampirkan kebijakan terkelola, lihat [Menambahkan izin identitas IAM \(konsol\) di Panduan Pengguna IAM](#).

Penebangan dan pemantauan di AWS CloudShell

Topik ini menjelaskan bagaimana Anda dapat mencatat dan memantau AWS CloudShell aktivitas dan kinerja dengan CloudTrail.

Memantau aktivitas dengan CloudTrail

AWS CloudShell terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS dalam AWS CloudShell. CloudTrail menangkap semua panggilan API untuk AWS CloudShell sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari AWS CloudShell konsol dan panggilan kode ke AWS CloudShell API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon Simple Storage Service (Amazon S3). Ini termasuk acara untuk AWS CloudShell.

Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menemukan berbagai informasi tentang permintaan. Misalnya, Anda dapat menentukan

permintaan yang dibuat ke AWS CloudShell, Anda dapat mempelajari alamat IP tempat permintaan dibuat, siapa yang membuat permintaan, dan kapan permintaan itu dibuat.

AWS CloudShell di CloudTrail

Tabel berikut mencantumkan AWS CloudShell peristiwa yang disimpan dalam file CloudTrail log.

Note

AWS CloudShell acara yang meliputi:

- *menunjukkan bahwa ini adalah panggilan API yang tidak bermutasi (hanya-baca).
- Kata tersebut `Environment` berkaitan dengan siklus hidup lingkungan komputasi yang menampung pengalaman shell.
- Kata `Layout` mengembalikan semua tab browser di terminal. CloudShell

CloudShell Acara di CloudTrail

Nama peristiwa	Deskripsi
<code>createEnvironment</code>	Terjadi ketika CloudShell lingkungan dibuat.
<code>createSession</code>	Terjadi ketika CloudShell lingkungan terhubung dari AWS Management Console.
<code>deleteEnvironment</code>	Terjadi ketika CloudShell lingkungan dihapus.
<code>deleteSession</code>	Terjadi ketika sesi di CloudShell tab yang berjalan di tab browser saat ini dihapus.
<code>getEnvironmentStatus*</code>	Terjadi ketika status CloudShell lingkungan diambil.
<code>getFileDownloadUrls*</code>	Terjadi ketika URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunduh file CloudShell melalui

Nama peristiwa	Deskripsi
	menggunakan antarmuka web CloudShell dihasilkan.
getFileUploadUrls*	Terjadi ketika URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunggah file CloudShell melalui menggunakan antarmuka web CloudShell dihasilkan.
getLayout*	Terjadi ketika CloudShell tata letak di awal sesi diambil.
putCredentials	Terjadi ketika kredensial yang digunakan untuk masuk ke AWS Management Console to CloudShell diteruskan.
redeemCode*	Terjadi ketika alur kerja untuk mengambil token penyegaran di CloudShell lingkungan dimulai. Anda nantinya dapat menggunakan token ini dalam <code>putCredentials</code> perintah untuk mengakses CloudShell lingkungan.
sendHeartBeat	Terjadi untuk mengkonfirmasi bahwa CloudShell sesi aktif.
startEnvironment	Terjadi ketika CloudShell lingkungan dimulai.
stopEnvironment	Terjadi ketika CloudShell lingkungan berjalan dihentikan.
updateLayout	Terjadi ketika tata letak saat ini dari aplikasi web di backend disimpan.

Acara yang menyertakan kata “Layout” mengembalikan semua tab browser di CloudShell terminal.

EventBridge aturan untuk AWS CloudShell tindakan

Dengan EventBridge aturan, Anda menentukan tindakan target yang akan diambil saat EventBridge menerima acara yang cocok dengan aturan. Anda dapat menentukan aturan yang menentukan tindakan target yang akan diambil berdasarkan AWS CloudShell tindakan yang direkam sebagai peristiwa dalam file CloudTrail log.

Misalnya, Anda dapat [membuat EventBridge aturan dengan AWS CLI](#) menggunakan `put-rule` perintah. `put-rule` Panggilan harus berisi setidaknya satu EventPattern atau ScheduleExpression. Aturan dengan EventPatterns dipicu ketika peristiwa yang cocok diamati. EventPattern Untuk AWS CloudShell acara:

```
{ "source": [ "aws.cloudshell" ], "detail-type": [ "AWS API Call via CloudTrail" ],
  "detail": { "eventSource": [ "cloudshell.amazonaws.com" ] } }
```

Untuk informasi selengkapnya, lihat [Peristiwa dan Pola Peristiwa EventBridge di Panduan EventBridge Pengguna Amazon](#).

Validasi kepatuhan untuk AWS CloudShell

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS layanan sebagai bagian dari beberapa program AWS kepatuhan.

AWS CloudShell berada dalam lingkup dengan program kepatuhan berikut:

SOC

AWS Laporan Sistem dan Kontrol Organisasi (SOC) adalah laporan pemeriksaan pihak ketiga independen yang menunjukkan bagaimana AWS mencapai kontrol dan tujuan kepatuhan utama.

Layanan	SDK	SOC 1,2,3
AWS CloudShell	CloudShell	✓

PCI

Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS) adalah standar keamanan informasi eksklusif yang dikelola oleh Dewan Standar Keamanan PCI, yang didirikan oleh American Express, Discover Financial Services, JCB International, Worldwide dan Visa Inc. MasterCard

Layanan	SDK	PCI
AWS CloudShell	CloudShell	✓

Sertifikasi dan Layanan ISO dan CSA STAR

AWS memiliki sertifikasi untuk kepatuhan dengan ISO/IEC 27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015, dan CSA STAR CCM v4.0.

Layanan	SDK	Sertifikasi dan Layanan ISO dan CSA STAR
AWS CloudShell	CloudShell	✓

FedRamp

Federal Risk and Authorization Management Program (FedRAMP) adalah program pemerintah AS yang memberikan pendekatan standar untuk penilaian keamanan, otorisasi, dan pemantauan berkelanjutan untuk produk dan layanan cloud.

Layanan	SDK	FedRAMP Moderat (Timur/Barat)	FedRAMP Tinggi (GovCloud)
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

The Department of Defense (DoD) Cloud Computing Security Requirements Guide (SRG) menyediakan penilaian standar dan proses otorisasi untuk penyedia layanan cloud (CSP) untuk mendapatkan otorisasi sementara DoD, sehingga mereka dapat melayani pelanggan DoD.

Layanan yang melalui penilaian dan otorisasi DoD CC SRG akan memiliki status sebagai berikut:

- Penilaian Organisasi Penilaian Pihak Ketiga (3PAO): Layanan ini saat ini sedang menjalani penilaian oleh penilai pihak ketiga kami.

- Tinjauan Joint Authorization Board (JAB): Layanan ini saat ini sedang menjalani tinjauan JAB.
- Tinjauan Badan Sistem Informasi Pertahanan (DISA): Layanan ini saat ini sedang menjalani tinjauan DISA.

Layanan	SDK	DoD CC SRG IL2 (Timur/Barat)	DoD CC SRG IL2 () GovCloud	DoD CC SRG IL4 () GovCloud	DoD CC SRG IL5 () GovCloud	DoD CC SRG IL6 (Wilayah Rahasia)AWS
AWS CloudShell	CloudShell	Penilaian 3PAO	N/A	N/A	N/A	N/A

HIPAA BAA

Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan 1996 (HIPAA) adalah undang-undang federal yang mewajibkan pembuatan standar nasional untuk melindungi informasi kesehatan pasien yang sensitif agar tidak diungkapkan tanpa persetujuan atau sepengetahuan pasien.

AWS memungkinkan entitas yang dilindungi dan rekan bisnis mereka yang tunduk pada HIPAA untuk memproses, menyimpan, dan mengirimkan informasi kesehatan yang dilindungi (PHI) dengan aman. Selain itu, pada Juli 2013, AWS menawarkan Adendum Asosiasi Bisnis standar (BAA) untuk pelanggan tersebut.

Layanan	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

Program Penilai Terdaftar Keamanan Informasi (IRAP) memungkinkan pelanggan Pemerintah Australia untuk memvalidasi bahwa kontrol yang sesuai telah ada dan menentukan model tanggung jawab yang sesuai untuk memenuhi persyaratan Manual Keamanan Informasi Pemerintah Australia (ISM) yang diproduksi oleh Australian Cyber Security Centre (ACSC).

Layanan	Ruang nama*	IRAP dilindungi
AWS CloudShell	N/A	✓

* Ruang nama membantu Anda mengidentifikasi layanan di seluruh lingkungan Anda. AWS Misalnya, saat Anda membuat kebijakan IAM, gunakan Amazon Resource Names (ARN), dan baca AWS CloudTrail log.

MTCS

Multi-Tier Cloud Security (MTCS) adalah Standar manajemen keamanan operasional Singapura (SPRING SS 584), berdasarkan standar ISO 27001/02 Information Security Management System (ISMS).

Layanan	SDK	AS-Timur (Ohio)	AS-Timur (Virginia Utara)	AS-Barat (Oregon)	AS-Barat (California Utara)	Singapura	Seoul
AWS CloudShell	CloudShell	✓	✓	✓	N/A	N/A	N/A

C5

Cloud Computing Compliance Controls Catalog (C5) adalah skema pengesahan yang didukung Pemerintah Jerman yang diperkenalkan di Jerman oleh Kantor Federal untuk Keamanan Informasi (BSI) untuk membantu organisasi menunjukkan keamanan operasional terhadap serangan cyber umum saat menggunakan layanan cloud dalam konteks “Rekomendasi Keamanan untuk Penyedia Cloud” Pemerintah Jerman.

Layanan	SDK	C5
AWS CloudShell	CloudShell	✓

ENS High

Skema akreditasi ENS (Esquema Nacional de Seguridad) telah dikembangkan oleh Kementerian Keuangan dan Administrasi Publik dan CCN (Pusat Kriptologi Nasional). Ini terdiri dari prinsip-prinsip dasar dan persyaratan minimum yang diperlukan untuk perlindungan informasi yang memadai.

Layanan	SDK	ENS Tinggi
AWS CloudShell	CloudShell	✓

FINMA

Otoritas Pengawas Pasar Keuangan Swiss (FINMA) adalah regulator pasar keuangan independen Swiss. AWS Keselarasan dengan persyaratan FINMA menunjukkan komitmen berkelanjutan kami untuk memenuhi harapan yang meningkat bagi penyedia layanan cloud yang ditetapkan oleh regulator dan pelanggan layanan keuangan Swiss.

Layanan	SDK	FINMA
AWS CloudShell	CloudShell	✓

PiTuKri

AWS Keselarasan dengan PiTuKri persyaratan menunjukkan komitmen berkelanjutan kami untuk memenuhi harapan yang meningkat untuk penyedia layanan cloud yang ditetapkan oleh Badan Transportasi dan Komunikasi Finlandia, Traficom.

Layanan	SDK	PiTuKri
AWS CloudShell	CloudShell	✓

Untuk daftar AWS layanan yang berada dalam cakupan program kepatuhan tertentu, lihat [AWS Services in Scope by Compliance Program](#) . Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga dengan menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Pengunduhan Laporan dalam AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS CloudShell ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan [Memulai Cepat Keamanan dan Kepatuhan Panduan Memulai](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan berfokus pada kepatuhan. AWS
- [Arsitektur untuk Whitepaper Keamanan dan Kepatuhan HIPAA — Whitepaper](#) ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Ketahanan di AWS CloudShell

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Selain infrastruktur AWS global, AWS CloudShell mendukung fitur khusus untuk mendukung ketahanan data dan kebutuhan cadangan Anda.

- Komit file yang Anda buat dan tambahkan AWS CodeCommit. Ini adalah layanan kontrol versi yang dihosting oleh Amazon Web Services yang dapat Anda gunakan untuk menyimpan dan mengelola aset secara pribadi di cloud. Aset ini dapat terdiri dari dokumen, kode sumber, dan file biner. Untuk informasi selengkapnya, lihat [Tutorial: Menggunakan CodeCommit diAWS CloudShell](#).
- Gunakan AWS CLI panggilan untuk menentukan file di direktori home Anda AWS CloudShell dan menemukannya sebagai objek di bucket Amazon S3. Sebagai contoh, lihat [tutorial memulai](#).

Keamanan infrastruktur di AWS CloudShell

Sebagai layanan terkelola, AWS CloudShell dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS CloudShell melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Note

Secara default, instal patch keamanan AWS CloudShell secara otomatis untuk paket sistem lingkungan komputasi Anda.

Analisis konfigurasi dan kerentanan di AWS CloudShell

Ini adalah tanggung jawab AWS CloudShell pengguna untuk memastikan bahwa perangkat lunak apa pun yang instal di lingkungan komputasi ditambah dan diperbarui.

Praktik terbaik keamanan untuk AWS CloudShell

Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan preskripsi.

Beberapa praktik terbaik keamanan untuk AWS CloudShell

- Gunakan izin dan kebijakan IAM untuk mengontrol akses ke AWS CloudShell dan memastikan pengguna hanya dapat melakukan tindakan tersebut (mengunduh dan mengunggah file, misalnya) yang diperlukan oleh peran mereka. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).
- Jangan sertakan data sensitif dalam entitas IAM Anda seperti pengguna, peran, atau nama sesi.
- Fitur Keep Safe Paste diaktifkan untuk menangkap potensi risiko keamanan dalam teks yang telah Anda salin dari sumber eksternal. Safe Paste diaktifkan secara default. Untuk informasi selengkapnya, lihat [Menggunakan Safe Paste untuk teks multiline](#).
- Biasakan dengan [Model Tanggung Jawab Keamanan Bersama](#) jika Anda menginstal aplikasi pihak ketiga ke lingkungan komputasi. AWS CloudShell
- Siapkan mekanisme rollback sebelum mengedit skrip shell yang memengaruhi pengalaman shell pengguna. Untuk informasi selengkapnya, lihat [Memodifikasi shell Anda dengan skrip](#).
- Simpan kode Anda dengan aman di sistem kontrol versi, misalnya, [AWS CodeCommit](#).

AWS CloudShell FAQ Keamanan

Jawaban atas pertanyaan yang sering diajukan tentang keamanan untuk ini Layanan AWS.

- [Apa saja AWS proses dan teknologi yang digunakan saat Anda meluncurkan CloudShell dan memulai sesi shell?](#)
- [Apakah mungkin untuk membatasi akses jaringan? CloudShell](#)
- [Dapatkah saya menyesuaikan CloudShell lingkungan saya?](#)
- [Di mana \\$HOME direktori saya sebenarnya disimpan di AWS Cloud?](#)
- [Apakah mungkin untuk mengenkripsi \\$HOME direktori saya?](#)

- [Bisakah saya menjalankan pemindaian virus di \\$HOME direktori saya?](#)

Apa saja AWS proses dan teknologi yang digunakan saat Anda meluncurkan CloudShell dan memulai sesi shell?

Saat masuk AWS Management Console, Anda memasukkan kredensial pengguna IAM Anda. Dan, saat Anda meluncurkan CloudShell dari antarmuka konsol, kredensial ini digunakan dalam panggilan ke CloudShell API yang membuat lingkungan komputasi untuk layanan tersebut. AWS Systems Manager Sesi kemudian dibuat untuk lingkungan komputasi, dan CloudShell mengirimkan perintah ke sesi itu.

[Kembali ke daftar FAQ keamanan](#)

Apakah mungkin untuk membatasi akses jaringan? CloudShell

Dimungkinkan untuk membatasi akses jaringan dengan menghubungkan CloudShell dengan penyedia jaringan Anda. Atau, Anda dapat menggunakan izin IAM, secara eksplisit menolak akses ke CloudShell atau tidak memberikan izin IAM apa pun, dan menggunakan fitur IAM penolakan implisit. Untuk informasi selengkapnya, lihat [Mengelola CloudShell akses dan penggunaan AWS dengan kebijakan IAM](#).

[Kembali ke daftar FAQ keamanan](#)

Dapatkah saya menyesuaikan CloudShell lingkungan saya?

Anda dapat mengunduh dan menginstal utilitas dan perangkat lunak pihak ketiga lainnya untuk CloudShell lingkungan Anda. Hanya perangkat lunak yang diinstal di \$HOME direktori Anda yang bertahan di antara sesi.

Sebagaimana didefinisikan oleh [model tanggung jawab AWS bersama](#), Anda bertanggung jawab atas konfigurasi dan pengelolaan aplikasi yang diperlukan yang Anda instal.

[Kembali ke daftar FAQ keamanan](#)

Di mana **\$HOME** direktori saya sebenarnya disimpan di AWS Cloud?

Infrastruktur untuk menyimpan data di Anda \$HOME disediakan oleh Amazon S3.

[Kembali ke daftar FAQ keamanan](#)

Apakah mungkin untuk mengenkripsi **\$HOME** direktori saya?

Data Anda di **\$HOME** direktori Anda sudah dienkripsi menggunakan Enkripsi Amazon S3.

[Kembali ke daftar FAQ keamanan](#)

Bisakah saya menjalankan pemindaian virus di **\$HOME** direktori saya?

Saat ini, tidak mungkin menjalankan pemindaian virus **\$HOME** direktori Anda. Support untuk fitur ini sedang ditinjau.

[Kembali ke daftar FAQ keamanan](#)

AWS CloudShell lingkungan komputasi: spesifikasi dan perangkat lunak

Saat Anda meluncurkan AWS CloudShell, lingkungan komputasi yang didasarkan pada [Amazon Linux 2023](#) dibuat untuk meng-host pengalaman shell. Lingkungan dikonfigurasi dengan [sumber daya komputasi \(vCPU dan memori\)](#) dan menyediakan berbagai perangkat lunak [pra-instal](#) yang dapat diakses dari antarmuka baris perintah. Anda juga dapat mengonfigurasi lingkungan default Anda dengan menginstal perangkat lunak dan memodifikasi skrip shell.

Menghitung sumber daya lingkungan

Setiap lingkungan AWS CloudShell komputasi diberi CPU dan sumber daya memori berikut:

- 1 vCPU (unit pemrosesan pusat virtual)
- RAM 2-GiB

Dan, lingkungan disediakan dengan konfigurasi penyimpanan berikut:

- Penyimpanan persisten 1-GB (penyimpanan tetap ada setelah sesi berakhir)

Untuk informasi selengkapnya, lihat [Penyimpanan tetap](#).

CloudShell persyaratan jaringan

WebSockets

CloudShell tergantung pada WebSocket protokol, yang memungkinkan komunikasi interaktif dua arah antara browser web pengguna dan CloudShell layanan di AWS Cloud. Jika Anda menggunakan browser di jaringan pribadi, akses aman ke internet mungkin difasilitasi oleh server proxy dan firewall. WebSocket Komunikasi biasanya dapat melintasi server proxy tanpa masalah. Tetapi dalam beberapa kasus, server proxy WebSockets mencegah bekerja dengan benar. Jika masalah ini terjadi, CloudShell antarmuka Anda melaporkan kesalahan berikut: `Failed to open sessions : Timed out while opening the session.`

Jika kesalahan ini terjadi berulang kali, lihat dokumentasi untuk server proxy Anda untuk memastikan bahwa itu dikonfigurasi untuk mengizinkan WebSockets. Atau, Anda dapat menghubungi administrator sistem jaringan Anda.

Note

Jika Anda ingin menentukan izin granular dengan mengizinkan daftar URL tertentu, Anda dapat menambahkan bagian dari URL yang digunakan AWS Systems Manager sesi untuk membuka WebSocket koneksi untuk mengirim input dan menerima output. (AWS CloudShellPerintah Anda dikirim ke sesi Systems Manager tersebut.)

Format untuk ini StreamUrl digunakan oleh Systems Manager adalah `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

Wilayah ini mewakili pengenal Wilayah untuk AWS Wilayah yang didukung oleh AWS Systems Manager, seperti `us-east-2` untuk Wilayah Timur AS (Ohio).

Karena `session-id` dibuat setelah sesi Systems Manager tertentu berhasil dimulai, Anda hanya dapat menentukan `wss://ssmmessages.region.amazonaws.com` saat memperbarui daftar izin URL Anda. Untuk informasi selengkapnya, lihat [StartSession](#) operasi di Referensi AWS Systems Manager API.

Perangkat lunak pra-instal

Note

Karena lingkungan AWS CloudShell pengembangan diperbarui secara berkala untuk menyediakan akses ke perangkat lunak terbaru, kami tidak memberikan nomor versi tertentu dalam dokumentasi ini. Sebagai gantinya, kami menjelaskan bagaimana Anda dapat memeriksa versi mana yang diinstal. Untuk memeriksa versi yang diinstal, masukkan nama program diikuti dengan `--version` opsi (misalnya, `git --version`).

Kerang

Cangkang yang sudah diinstal sebelumnya

Nama	Penjelasan	Informasi versi
Bash	Shell Bash adalah aplikasi shell default untuk AWS CloudShell.	<code>bash --version</code>
PowerShell (pwsh)	Menawarkan antarmuka baris perintah dan dukungan bahasa scripting, PowerShell dibangun di atas Microsoft .NET Command Language Runtime. PowerShell menggunakan perintah ringan cmdlets yang disebut yang menerima dan mengembalikan objek .NET.	<code>pwsh --version</code>
Z Shell (zsh)	Z Shell, juga dikenal sebagai zsh, adalah versi diperpanjang dari Bourne Shell yang menawarkan dukungan penyesuaian yang ditingkatkan untuk tema dan plugin.	<code>zsh --version</code>

AWS Antarmuka baris perintah (CLI)

CLI

Nama	Penjelasan	Informasi versi
AWS CDK CLI Toolkit	AWS CDK Toolkit, perintah CLI <code>cdk</code> , adalah alat utama yang berinteraksi dengan	<code>cdk --version</code>

Nama	Penjelasan	Informasi versi
	<p>aplikasi Anda. AWS CDK Ini mengeksekusi aplikasi Anda, menginterogasi model aplikasi yang Anda tetapkan, dan menghasilkan serta menerapkan AWS CloudFormation template yang dihasilkan oleh. AWS CDK</p> <p>Untuk informasi selengkapnya, lihat AWS CDK Toolkit.</p>	
AWS CLI	<p>AWS CLI ini adalah antarmuka baris perintah yang dapat Anda gunakan untuk mengelola beberapa AWS layanan dari baris perintah dan mengotomatiskannya menggunakan skrip. Untuk informasi selengkapnya, lihat Bekerja dengan AWS layanan di AWS CloudShell.</p> <p>Untuk informasi tentang cara memastikan bahwa Anda menggunakan up-to-date versi paling banyak AWS CLI versi 2, lihat Instalasi AWS CLI ke direktori home Anda.</p>	<pre>aws --version</pre>

Nama	Penjelasan	Informasi versi
EB CLI	<p>AWS Elastic Beanstalk CLI menyediakan antarmuka baris perintah untuk menyederhanakan pembuatan, pembaruan, dan pemantauan lingkungan dari repositori lokal.</p> <p>Untuk informasi selengkapnya, lihat Menggunakan antarmuka baris perintah Elastic Beanstalk (EB CLI) di Panduan Pengembang. AWS Elastic Beanstalk</p>	<code>eb --version</code>
Amazon ECS CLI	<p>Antarmuka baris perintah Amazon Elastic Container Service (Amazon ECS) (CLI) menyediakan perintah tingkat tinggi untuk menyederhanakan pembuatan, pembaruan, dan pemantauan cluster dan tugas.</p> <p>Untuk informasi selengkapnya, lihat Menggunakan Antarmuka Baris Perintah Amazon ECS di Panduan Pengembang Layanan Amazon Elastic Container.</p>	<code>ecs-cli --version</code>

Nama	Penjelasan	Informasi versi
CLI AWS SAM	<p>AWS SAMCLI adalah alat baris perintah yang beroperasi pada AWS Serverless Application Model template dan kode aplikasi. Anda dapat melakukan beberapa tugas. Ini termasuk menjalankan fungsi Lambda secara lokal, membuat paket penerapan untuk aplikasi tanpa server Anda, dan menerapkan aplikasi tanpa server Anda ke Cloud. AWS</p> <p>Untuk informasi selengkapnya, lihat referensi perintah AWS SAM CLI di Panduan AWS Serverless Application Model Pengembang.</p>	<code>sam --version</code>

Nama	Penjelasan	Informasi versi
AWS Tools for PowerShell	<p>AWS Tools for PowerShell ini adalah PowerShell modul yang dibangun di atas fungsionalitas yang diekspos oleh AWS SDK for .NET. Dengan AWS Tools for PowerShell, Anda dapat membuat skrip operasi pada AWS sumber daya Anda dari baris PowerShell perintah.</p> <p>AWS CloudShell pra-instal versi termodulasi (AWS.tools) dari AWS Tools for PowerShell</p> <p>Untuk informasi selengkapnya, lihat Menggunakan Alat AWS untuk PowerShell di Panduan AWS Tools for PowerShell Pengguna.</p>	<pre>powershell --Command 'Get-Module -ListAvailable -Name AWS.Tools.Common'</pre>

Runtime dan AWS SDK: Node.js dan Python 3

Runtime dan AWS SDK

Nama	Penjelasan	Informasi versi
Node.js (dengan npm)	<p>Node.js adalah JavaScript runtime yang dirancang untuk memudahkan penerapan teknik pemrograman asinkron. Untuk informasi selengkapnya, lihat dokumentasi di situs resmi Node.js.</p>	<ul style="list-style-type: none"> • Node.js: <code>node --version</code> • npm: <code>npm --version</code>

Nama	Penjelasan	Informasi versi
	<p>npm adalah manajer paket yang menyediakan akses ke registri JavaScript modul online. Untuk informasi lebih lanjut, lihat dokumentasi di situs resmi npm.</p>	
SDK untuk JavaScript di Node.js	<p>Kit pengembangan perangkat lunak (SDK) membantu menyederhanakan pengembangan dengan menyediakan JavaScript objek untuk layanan AWS termasuk Amazon S3, Amazon EC2, DynamoDB, dan Amazon SWF. Untuk informasi selengkapnya, lihat Panduan Developer AWS SDK for JavaScript.</p>	<pre>npm -g ls --depth 0 2>/dev/null grep aws-sdk</pre>

Nama	Penjelasan	Informasi versi
Python	<p>Python 3 siap digunakan di lingkungan shell. Python 3 sekarang dianggap sebagai versi default dari bahasa pemrograman (dukungan untuk Python 2 berakhir pada Januari 2020). Untuk informasi selengkapnya, lihat dokumentasi di situs resmi Python.</p> <p>Juga, pra-instal adalah pip, penginstal paket untuk Python. Anda dapat menggunakan program baris perintah ini untuk menginstal paket Python dari indeks online seperti Indeks Paket Python. Untuk informasi selengkapnya, lihat dokumentasi yang disediakan oleh Otoritas Kemasan Python.</p>	<ul style="list-style-type: none">• Python 3: <code>python3 --version</code>• pip: <code>pip3 --version</code>

Nama	Penjelasan	Informasi versi
SDK untuk Python (Boto3)	<p>Boto adalah perangkat pengembangan perangkat lunak (SDK) yang digunakan pengembang Python untuk membuat, mengkonfigurasi, dan mengelola Layanan AWS, seperti Amazon EC2 dan Amazon S3. SDK menyediakan API berorientasi objek easy-to-use, serta akses tingkat rendah ke Layanan AWS</p> <p>Untuk informasi lebih lanjut, lihat dokumentasi Boto3.</p>	<pre>pip3 list grep boto3</pre>

Alat pengembangan dan utilitas shell

Alat pengembangan dan utilitas shell

Nama	Penjelasan	Informasi versi
penyelesaian bash-	<p>bash-completion adalah kumpulan fungsi shell yang memungkinkan pelengkapan otomatis perintah atau argumen yang diketik sebagian dengan menekan tombol Tab. Anda dapat menemukan paket yang didukung bash-completion.</p> <pre>/usr/share/bash-completion/completions</pre>	<pre>dnf info bash-completion</pre>

Nama	Penjelasan	Informasi versi
	<p>Untuk mengatur pelengkapan otomatis untuk perintah paket, file program harus bersumber . Misalnya, untuk menyiapkan pelengkapan otomatis untuk perintah Git, tambahkan baris berikut .bashrc agar fitur tersedia setiap kali AWS CloudShell sesi Anda dimulai:</p> <pre>source /usr/share/ bash-completion/ completions/git</pre> <p>Jika Anda ingin menggunakan skrip penyelesaian kustom, tambahkan ke direktori home persisten (\$HOME) dan sumber langsung .bashrc</p> <p>Untuk informasi selengkapnya, lihat halaman README proyek di GitHub.</p>	

Nama	Penjelasan	Informasi versi
CodeCommit utilitas untuk Git	<p>git-remote-codecommit adalah utilitas yang menyediakan metode sederhana untuk mendorong dan menarik kode dari CodeCommit repositori dengan Git. Ini adalah metode yang direkomendasikan untuk mendukung koneksi yang dibuat dengan akses federasi, penyedia identitas, dan kredensial sementara.</p> <p>Untuk informasi selengkapnya, lihat Langkah persiapan untuk koneksi HTTPS ke AWS CodeCommit dengan git-remote-codecommit di Panduan AWS CodeCommit Pengguna.</p>	<pre>pip3 list grep git-remote-codecommit</pre>
Git	<p>Git adalah sistem kontrol versi terdistribusi yang mendukung praktik pengembangan perangkat lunak modern melalui alur kerja cabang dan pementasan konten. Untuk informasi selengkapnya, lihat halaman dokumentasi di situs resmi Git.</p>	<pre>git --version</pre>
iputil	<p>Paket iputils berisi utilitas untuk jaringan Linux. Untuk informasi lebih lanjut tentang utilitas yang disediakan, lihat repositori iputils di GitHub</p>	<p>Contoh untuk alat iputils:</p> <pre>arping -V</pre>

Nama	Penjelasan	Informasi versi
jq	Utilitas jq mem-parsing data berformat JSON untuk menghasilkan output yang dimodifikasi oleh filter baris perintah. Untuk informasi lebih lanjut, lihat manual jq yang dihosting di GitHub .	jq --version
kubectl	kubectl adalah alat baris perintah untuk berkomunikasi dengan control plane kluster Kubernetes, menggunakan API Kubernetes.	kubectl --version
membuat	Utilitas make digunakan <code>makefiles</code> untuk mengotomatiskan set tugas dan mengatur kompilasi kode. Untuk informasi selengkapnya, lihat dokumentasi GNU Make .	make --version
pria	Perintah <code>man</code> menyediakan halaman manual untuk utilitas dan alat baris perintah. Misalnya, <code>man ls</code> mengembalikan halaman manual untuk <code>ls</code> perintah yang mencantumkan isi direktori. Untuk informasi lebih lanjut, lihat entri Wikipedia di halaman manual .	man --version

Nama	Penjelasan	Informasi versi
nano	nano adalah editor kecil dan user-friendly untuk antarmuka berbasis teks. Untuk informasi lebih lanjut, lihat dokumentasi GNU nano .	nano --version
procs	procs adalah utilitas administrasi sistem yang dapat Anda gunakan untuk memantau dan menghentikan proses yang sedang berjalan. Untuk informasi selengkapnya, lihat file README yang mencantumkan program yang dapat dijalankan dengan procs .	ps --version
Klien SSH	Klien SSH menggunakan protokol shell aman untuk komunikasi terenkripsi dengan komputer jarak jauh. OpenSSH adalah klien SSH yang sudah diinstal sebelumnya. Untuk informasi selengkapnya, lihat situs OpenSSH yang dikelola oleh OpenBSD .	ssh -V

Nama	Penjelasan	Informasi versi
sudo	Dengan utilitas sudo, pengguna dapat menjalankan program dengan izin keamanan pengguna lain, biasanya superuser. Sudo berguna ketika Anda perlu menginstal aplikasi sebagai administrator sistem. Untuk informasi lebih lanjut, lihat Manual Sudo .	sudo --version
tar	tar adalah utilitas baris perintah yang dapat Anda gunakan untuk mengelompokkan beberapa file dalam satu file arsip (sering disebut tarball). Untuk informasi lebih lanjut, lihat dokumentasi tar GNU .	tar --version
tmux	tmux adalah multiplexer terminal yang dapat Anda gunakan untuk menjalankan berbagai program secara simultan di beberapa jendela. Untuk informasi lebih lanjut, lihat blog yang menyediakan pengantar singkat untuk tmux .	tmux -V
unzip	Untuk informasi selengkapnya, lihat zip/unzip .	

Nama	Penjelasan	Informasi versi
vim	vim adalah editor yang dapat disesuaikan yang dapat Anda berinteraksi melalui antarmuka berbasis teks. Untuk informasi selengkapnya, lihat sumber dokumentasi yang disediakan di vim.org .	<code>vim --version</code>
wget	wget adalah program komputer yang digunakan untuk mengambil konten dari server web yang ditentukan oleh titik akhir di baris perintah. Untuk informasi lebih lanjut, lihat dokumentasi GNU Wget .	<code>wget --version</code>
zip/unzip	Utilitas zip/unzip menggunakan format file arsip yang memberikan kompresi data lossless tanpa kehilangan data. Panggil perintah zip untuk mengelompokkan dan mengompres file dalam satu arsip. Gunakan unzip untuk mengekstrak file dari arsip ke direktori tertentu.	<code>unzip --version</code> <code>zip --version</code>

Nama	Penjelasan	Informasi versi
Docker	<p>Docker adalah platform terbuka untuk mengembankan, mengirim, dan menjalankan aplikasi. Docker memungkinkan Anda untuk memisahkan aplikasi Anda dari infrastruktur Anda sehingga Anda dapat mengirimkan perangkat lunak dengan cepat. Ini memungkinkan Anda untuk membangun Dockerfiles di dalam AWS CloudShell, dan membangun aset Docker dengan CDK. Untuk informasi tentang Wilayah mana yang didukung dengan Docker, lihat Wilayah Docker. Anda harus menyadari bahwa Docker memiliki ruang terbatas di lingkungan. Jika Anda memiliki gambar individual yang besar, atau terlalu banyak gambar Docker yang sudah ada sebelumnya, itu dapat menyebabkan masalah. Untuk informasi selengkapnya tentang Docker, lihat panduan Dokumentasi Docker.</p>	<code>docker --version</code>

Instalasi AWS CLI ke direktori home Anda

Seperti perangkat lunak lainnya yang sudah diinstal sebelumnya di CloudShell lingkungan Anda, AWS CLI alat ini diperbarui secara otomatis dengan peningkatan terjadwal dan tambalan keamanan.

Jika Anda ingin memastikan bahwa Anda memiliki up-to-date versi terbanyak AWS CLI, Anda dapat memilih untuk menginstal alat secara manual di direktori home shell.

Important

Anda perlu menginstal salinan Anda secara manual AWS CLI di direktori home sehingga tersedia saat berikutnya Anda memulai CloudShell sesi. Instalasi ini diperlukan karena file yang ditambahkan ke direktori di luar \$HOME dihapus setelah Anda menyelesaikan sesi shell. Juga, setelah Anda menginstal salinan ini AWS CLI, itu tidak diperbarui secara otomatis. Dengan kata lain, Anda bertanggung jawab untuk mengelola pembaruan dan patch keamanan.

Untuk informasi selengkapnya tentang Model Tanggung Jawab AWS Bersama, lihat [Perlindungan data di AWS CloudShell](#).

Untuk menginstal AWS CLI

1. Di baris CloudShell perintah, gunakan `curl` perintah untuk mentransfer salinan zip yang AWS CLI diinstal ke shell:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. Buka zip folder zip:

```
unzip awscliv2.zip
```

3. Untuk menambahkan alat ke folder tertentu, jalankan AWS CLI installer:

```
sudo ./aws/install --install-dir /home/cloudshell-user/usr/local/aws-cli --bin-dir /home/cloudshell-user/usr/local/bin
```

Jika berhasil diinstal, baris perintah menampilkan pesan berikut:

```
You can now run: /home/cloudshell-user/usr/local/bin/aws --version
```

4. Untuk kenyamanan Anda sendiri, kami sarankan Anda juga memperbarui variabel PATH lingkungan sehingga Anda tidak perlu menentukan jalur ke instalasi alat saat menjalankan aws perintah:


```
export PATH=/home/cloudshell-user/usr/local/bin:$PATH
```

Note

Jika Anda membatalkan perubahan iniPATH, aws perintah yang tidak menampilkan jalur tertentu menggunakan versi pra-instal secara defaultAWS CLI.

Menginstal perangkat lunak pihak ketiga di lingkungan shell Anda

Note

Kami menyarankan Anda meninjau [Model Tanggung Jawab Keamanan Bersama](#) sebelum Anda menginstal aplikasi pihak ketiga ke AWS CloudShell lingkungan komputasi.

Secara default, semua AWS CloudShell pengguna memiliki izin sudo. Oleh karena itu, Anda dapat menggunakan sudo perintah untuk menginstal perangkat lunak yang belum tersedia di lingkungan komputasi shell. Misalnya, Anda dapat menggunakan sudo dengan utilitas manajemen paket DNF untuk menginstal cowsay, yang menghasilkan gambar seni ASCII sapi dengan pesan:

```
sudo dnf install cowsay
```

Anda kemudian dapat meluncurkan program yang baru diinstal dengan mengetikecho "Welcome to AWS CloudShell" | cowsay.

Important

Package mengelola utilitas seperti dnf install program di direktori /usr/bin (, misalnya), yang didaur ulang ketika sesi shell Anda berakhir. Ini berarti perangkat lunak tambahan diinstal dan digunakan berdasarkan per sesi.

Memodifikasi shell Anda dengan skrip

Jika Anda ingin memodifikasi lingkungan shell default, Anda dapat mengedit skrip shell yang berjalan setiap kali lingkungan shell dimulai. `.bashrc` Skrip berjalan setiap kali shell bash default dimulai.

Warning

Jika Anda salah memodifikasi `.bashrc` file Anda, Anda mungkin tidak dapat mengakses lingkungan shell Anda sesudahnya. Ini praktik yang baik untuk membuat salinan file sebelum mengedit. Anda juga dapat mengurangi risiko dengan membuka dua shell saat mengedit. `.bashrc` Jika Anda kehilangan akses di satu shell, Anda masih masuk ke shell lain dan dapat memutar kembali perubahan apa pun.

Jika Anda kehilangan akses setelah salah memodifikasi `.bashrc` atau file lainnya, Anda dapat kembali AWS CloudShell ke pengaturan default dengan [menghapus direktori home Anda](#).

Dalam prosedurnya, Anda akan memodifikasi `.bashrc` skrip sehingga lingkungan shell Anda beralih secara otomatis untuk menjalankan shell Z.

1. Buka `.bashrc` menggunakan editor teks (Vim, misalnya):

```
vim .bashrc
```

2. Di antarmuka editor, tekan tombol I untuk mulai mengedit, lalu tambahkan yang berikut ini:

```
zsh
```

3. Untuk keluar dan menyimpan `.bashrc` file yang diedit, tekan Esc untuk masuk ke mode perintah Vim dan masukkan yang berikut ini:

```
:wq
```

4. Gunakan `source` perintah untuk memuat ulang `.bashrc` file:

```
source .bashrc
```

Ketika antarmuka baris perintah tersedia lagi, simbol prompt telah berubah `%` untuk menunjukkan bahwa Anda sekarang menggunakan shell Z.

AWS CloudShell bermigrasi dari Amazon Linux 2 ke Amazon Linux 2023

AWS CloudShell, yang didasarkan pada Amazon Linux 2 (AL2), telah bermigrasi ke Amazon Linux 2023 (AL2023). Untuk informasi selengkapnya tentang AL2023, lihat [Apa itu Amazon Linux 2023 \(AL2023\) di Panduan Pengguna Amazon Linux 2023](#).

Dengan AL2023, Anda dapat terus mengakses CloudShell lingkungan yang ada dengan semua alat yang disediakan oleh CloudShell. Untuk informasi selengkapnya tentang alat yang tersedia, lihat [Perangkat lunak pra-instal](#).

AL2023 menyediakan beberapa perbaikan pada alat pengembangan, termasuk versi paket yang lebih baru Node seperti.js 18 dan 3.9. Python

Note

Di AL2023, Python 2 tidak lagi dikirim bersama lingkungan Anda. CloudShell

Untuk informasi selengkapnya tentang perbedaan utama antara AL2 dan AL2023, lihat [Membandingkan Amazon Linux 2 dan Amazon Linux 2023 di Panduan Pengguna Amazon Linux 2023](#).

Jika Anda memiliki pertanyaan, hubungi [AWS Support](#). Anda juga dapat mencari jawaban dan memposting pertanyaan di [AWS re:Post](#). Saat masuk AWS re:Post, Anda mungkin diminta untuk masuk AWS.

AWS CloudShell FAQ Migrasi

Berikut ini adalah jawaban atas beberapa pertanyaan umum tentang migrasi dari AL2 ke AL2023 dengan AWS CloudShell

- [Apakah migrasi ini akan memengaruhi AWS sumber daya saya yang lain, seperti instans Amazon EC2 yang berjalan di AL2?](#)
- [Paket apa yang akan diubah dengan migrasi ke AL2023?](#)
- [Dapatkah saya memilih keluar dari migrasi?](#)
- [Bisakah saya membuat cadangan AWS CloudShell lingkungan saya?](#)

Apakah migrasi ini akan memengaruhi AWS sumber daya saya yang lain, seperti instans Amazon EC2 yang berjalan di AL2?

Tidak ada layanan atau sumber daya selain AWS CloudShell lingkungan Anda yang terpengaruh oleh migrasi ini. Ini termasuk sumber daya yang mungkin telah Anda buat atau akses dari dalam AWS CloudShell. Misalnya, jika Anda telah membuat instans Amazon EC2 yang berjalan di AL2, ini tidak akan dimigrasikan ke AL2023.

Paket apa saja yang telah diubah dengan migrasi ke AL2023?

AWS CloudShell lingkungan saat ini termasuk perangkat lunak pra-instal. Untuk mempelajari tentang daftar lengkap perangkat lunak pra-instal, lihat Perangkat lunak [pra-instal](#). AWS CloudShell akan terus mengirimkan paket-paket ini, dengan pengecualian Python 2. Untuk perbedaan lengkap antara paket yang disediakan oleh AL2 dan AL2023, lihat [Membandingkan AL2 dan AL2023](#). Untuk pelanggan dengan persyaratan paket dan versi tertentu yang tidak lagi terpenuhi setelah migrasi ke AL2023, kami sarankan menghubungi AWS Support untuk mengirimkan permintaan.

Dapatkah saya memilih keluar dari migrasi

Jawabannya adalah TIDAK. AWS CloudShell lingkungan dikelola oleh AWS, oleh karena itu, semua lingkungan telah ditingkatkan ke AL2023.

Bisakah saya membuat cadangan AWS CloudShell lingkungan saya?

AWS CloudShell akan terus mempertahankan direktori home pengguna. Untuk informasi selengkapnya, lihat [Kuota dan batasan layanan untuk AWS CloudShell](#). Jika Anda memiliki file atau konfigurasi yang disimpan di folder rumah Anda dan jika Anda ingin membuat cadangan untuk hal yang sama, selesaikan [Langkah 6: Buat cadangan direktori home](#).

Pemecahan Masalah AWS CloudShell

Saat menggunakan AWS CloudShell, Anda mungkin mengalami masalah, seperti ketika Anda meluncurkan CloudShell atau melakukan tugas-tugas utama menggunakan antarmuka baris perintah shell. Informasi yang tercakup dalam Bab ini mencakup cara memecahkan masalah beberapa masalah umum yang mungkin Anda temui.

Untuk jawaban atas berbagai pertanyaan tentang CloudShell, lihat [AWS CloudShellFAQ](#). Anda juga dapat mencari jawaban dan memposting pertanyaan di [Forum AWS CloudShell Diskusi](#). Ketika Anda memasuki forum ini, Anda mungkin diminta untuk masuk AWS. Anda juga dapat [kontak kami](#) secara langsung.

Memecahkan masalah kesalahan

Ketika Anda menemukan salah satu kesalahan terindeks berikut, Anda dapat menggunakan solusi berikut untuk mengatasi kesalahan ini.

Topik

- [Tidak dapat memulai lingkungan. Untuk mencoba lagi, refresh browser atau restart dengan memilih Actions, Restart AWS CloudShell](#)
- [Tidak dapat memulai lingkungan. Anda tidak memiliki izin yang diperlukan. Minta administrator IAM Anda untuk memberikan akses ke AWS CloudShell](#)
- [Tidak dapat mengakses AWS CloudShell baris perintah](#)
- [Tidak dapat melakukan ping ke alamat IP eksternal](#)
- [Ada beberapa masalah dalam mempersiapkan terminal Anda](#)
- [Tombol panah tidak berfungsi dengan benar di PowerShell](#)
- [Soket Web yang tidak didukung menyebabkan kegagalan untuk memulai sesi CloudShell](#)
- [Tidak dapat mengimpor AWSPowerShell.NetCore modul](#)
- [Docker tidak berjalan saat menggunakan AWS CloudShell](#)
- [Docker kehabisan ruang disk](#)
- [docker push waktunya habis dan terus mencoba lagi](#)

Tidak dapat memulai lingkungan. Untuk mencoba lagi, refresh browser atau restart dengan memilih Actions, Restart AWS CloudShell

Masalah: Ketika Anda mencoba untuk memulai AWS CloudShell dari AWS Management Console, Anda ditolak akses bahkan setelah Anda memiliki izin yang diperlukan dari administrator IAM Anda dan Anda telah menyegarkan browser Anda atau memulai ulang. CloudShell

Solusi: Hubungi [AWS Support](#).

[\(kembali ke atas\)](#)

Tidak dapat memulai lingkungan. Anda tidak memiliki izin yang diperlukan. Minta administrator IAM Anda untuk memberikan akses ke AWS CloudShell

Masalah: Saat Anda mencoba meluncurkan AWS CloudShell dari AWS Management Console, Anda ditolak aksesnya dan diberi tahu bahwa Anda tidak memiliki izin yang diperlukan.

Penyebab: Identitas IAM yang Anda gunakan untuk mengakses AWS CloudShell tidak memiliki izin IAM yang diperlukan.

Solusi: Minta administrator IAM Anda untuk memberi Anda izin yang diperlukan. Mereka dapat melakukan ini baik dengan menambahkan kebijakan AWS terkelola terlampir (AWS CloudShellFullAccess) atau kebijakan inline yang disematkan. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

[\(kembali ke atas\)](#)

Tidak dapat mengakses AWS CloudShell baris perintah

Masalah: Setelah memodifikasi file yang digunakan lingkungan komputasi, Anda tidak dapat mengakses baris perintah. AWS CloudShell

Solusi: Jika Anda kehilangan akses setelah salah memodifikasi `.bashrc` atau file lainnya, Anda dapat kembali AWS CloudShell ke pengaturan default dengan [menghapus direktori home Anda](#).

[\(kembali ke atas\)](#)

Tidak dapat melakukan ping ke alamat IP eksternal

Masalah: Ketika Anda menjalankan perintah ping dari baris perintah (misalnya, `ping amazon.com`), Anda menerima pesan berikut.

```
ping: socket: Operation not permitted
```

Penyebab: Utilitas ping menggunakan Internet Control Message Protocol (ICMP) untuk mengirim paket permintaan gema ke host target. Ia menunggu gema untuk membalas dari target. Karena protokol ICMP tidak diaktifkan AWS CloudShell, utilitas ping tidak beroperasi di lingkungan komputasi shell.

Solusi: Karena ICMP tidak didukung AWS CloudShell, Anda dapat menjalankan perintah berikut untuk menginstal Netcat. Netcat adalah utilitas jaringan komputer untuk membaca dari, dan menulis ke, koneksi jaringan menggunakan TCP atau UDP.

```
sudo yum install nc
nc -zv www.amazon.com 443
```

[\(kembali ke atas\)](#)

Ada beberapa masalah dalam mempersiapkan terminal Anda

Masalah: Saat mencoba mengakses AWS CloudShell menggunakan browser Microsoft Edge, Anda tidak dapat memulai sesi shell, dan browser menampilkan pesan kesalahan.

Penyebab: AWS CloudShell tidak kompatibel dengan versi Microsoft Edge sebelumnya. Anda dapat mengakses AWS CloudShell menggunakan empat versi utama terbaru dari [browser yang didukung](#).

Solusi: Instal versi terbaru browser Edge dari [situs Microsoft](#).

[\(kembali ke atas\)](#)

Tombol panah tidak berfungsi dengan benar di PowerShell

Masalah: Dalam operasi normal, Anda dapat menggunakan tombol panah untuk menavigasi antarmuka baris perintah dan memindai mundur dan maju melalui riwayat perintah Anda. Tapi, ketika Anda menekan tombol panah dalam versi tertentu dari PowerShell on AWS CloudShell, huruf mungkin salah dikeluarkan.

Penyebab: Situasi di mana tombol panah salah menampilkan huruf adalah masalah yang diketahui dengan versi PowerShell 7.2.x yang berjalan di Linux.

Solusi: Untuk menghapus urutan escape yang mengubah perilaku tombol panah, edit file PowerShell profil dan atur `$PSStyle` variabel ke `PlainText`.

1. Di baris AWS CloudShell perintah, masukkan perintah berikut untuk membuka file profil.

```
vim ~/.config/powershell/Microsoft.PowerShell_profile.ps1
```

Note

Jika Anda sudah masuk PowerShell, Anda juga dapat membuka file profil di editor dengan perintah berikut.

```
vim $PROFILE
```

2. Di editor, pergi ke akhir teks file yang ada, tekan i untuk masuk ke mode Sisipkan, dan kemudian tambahkan pernyataan berikut.

```
$PSStyle.OutputRendering = 'PlainText'
```

3. Setelah Anda melakukan pengeditan, tekan Esc untuk masuk ke mode perintah. Selanjutnya, masukkan perintah berikut untuk menyimpan file dan keluar dari editor.

```
:wq
```

Note

Perubahan Anda akan berlaku saat berikutnya Anda memulai PowerShell.

[\(kembali ke atas\)](#)

Soket Web yang tidak didukung menyebabkan kegagalan untuk memulai sesi CloudShell

Masalah: Saat mencoba memulai AWS CloudShell, Anda berulang kali menerima pesan berikut: `Failed to open sessions : Timed out while opening the session.`

Penyebab: CloudShell tergantung pada WebSocket protokol, yang memungkinkan komunikasi interaktif dua arah antara browser web Anda dan AWS CloudShell. Jika Anda menggunakan browser di jaringan pribadi, akses aman ke internet mungkin difasilitasi oleh server proxy dan firewall.

WebSocket Komunikasi biasanya dapat melintasi server proxy tanpa masalah. Namun, dalam beberapa kasus, server proxy WebSockets mencegah agar tidak berfungsi dengan benar. Jika masalah ini terjadi, tidak CloudShell dapat memulai sesi shell dan upaya untuk terhubung akhirnya habis.

Solusi: Batas waktu koneksi mungkin disebabkan oleh masalah selain tidak WebSockets didukung. Jika ini masalahnya, pertama-tama segarkan jendela browser tempat antarmuka baris CloudShell perintah berada.

Jika Anda masih mendapatkan kesalahan batas waktu setelah penyegaran, lihat dokumentasi untuk server proxy Anda. Dan, pastikan bahwa server proxy Anda dikonfigurasi untuk mengizinkan Web Sockets. Atau, hubungi administrator sistem jaringan Anda.

Note

Katakanlah Anda ingin menentukan izin granular dengan mengizinkan daftar URL tertentu. Anda dapat menambahkan bagian dari URL yang digunakan AWS Systems Manager sesi untuk membuka WebSocket koneksi untuk mengirim input dan menerima output. AWS CloudShellPerintah Anda dikirim ke sesi Systems Manager tersebut.

Format untuk ini StreamUrl yang digunakan oleh Systems Manager adalah `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

Wilayah mewakili pengenal Wilayah untuk Wilayah AWS yang didukung oleh AWS Systems Manager. Misalnya, `us-east-2` adalah pengenal Wilayah untuk Wilayah Timur AS (Ohio). Karena `session-id` dibuat setelah sesi Systems Manager tertentu berhasil dimulai, Anda hanya dapat menentukan `wss://ssmmessages.region.amazonaws.com` kapan Anda memperbarui daftar izin URL Anda. Untuk informasi selengkapnya, lihat [StartSession](#) operasi di Referensi AWS Systems Manager API.

[\(kembali ke atas\)](#)

Tidak dapat mengimpor **AWSPowerShell.NetCore** modul

Masalah: Saat Anda mengimpor file `AWSPowerShell.NetCore` modul di PowerShell by `Import-Module -Name AWSPowerShell.NetCore`, Anda menerima pesan galat berikut:

Import-Module: Modul yang ditentukan 'AWSPowerShell.NetCore' tidak dimuat karena tidak ada file modul yang valid ditemukan di direktori modul apa pun.

Penyebab: `AWSPowerShell.NetCore` Modul digantikan oleh modul `AWS.Tools` per layanan di `AWS CloudShell`

Solusi: Setiap pernyataan impor eksplisit mungkin tidak lagi diperlukan atau perlu diubah ke modul `.Tools` per layanan terkait `AWS`.

Example

Example

- Untuk kebanyakan kasus, selama tidak ada jenis `.Net` yang digunakan, Anda tidak memerlukan pernyataan impor eksplisit. Berikut ini adalah contoh pernyataan impor.
 - `Get-S3Bucket`
 - `(Get-EC2Instance).Instances`
- Jika jenis `.Net` digunakan, impor modul tingkat layanan (`()AWS.Tools.<Service>`). Berikut ini adalah contoh sintaks.

```
Import-Module -Name AWS.Tools.EC2
$instanceTag = [Amazon.EC2.Model.Tag]::new("Environment","Dev")
```

```
Import-Module -Name AWS.Tools.S3
$lifecycleRule = [Amazon.S3.Model.LifecycleRule]::new()
```

Untuk informasi lebih lanjut, lihat [pengumuman versi 4](#) untuk `AWS Tools for PowerShell`.

[\(kembali ke atas\)](#)

Docker tidak berjalan saat menggunakan `AWS CloudShell`

Masalah: `Docker` tidak berjalan dengan baik saat menggunakan `AWS CloudShell`. Anda menerima pesan galat berikut: `docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?.`

Solusi: Coba mulai ulang lingkungan Anda. Pesan kesalahan ini dapat terjadi ketika Anda menjalankan `Docker` `AWS CloudShell` di Wilayah yang tidak mendukungnya. Pastikan Anda menjalankan `Docker` di Region yang didukung, untuk informasi tentang Wilayah mana yang mendukung penggunaan container `Docker` `AWS CloudShell`, lihat [Docker Regions](#).

Docker kehabisan ruang disk

Masalah: Anda menerima pesan kesalahan berikut: `ERROR: failed to solve: failed to register layer: write [...]: no space left on device.`

Penyebab: Dockerfile melebihi ruang disk yang tersedia di AWS CloudShell. Ini dapat disebabkan karena gambar individu yang besar atau terlalu banyak gambar Docker yang sudah ada sebelumnya.

Solusi: Jalankan `df -h` untuk menemukan penggunaan disk. Jalankan `sudo du -sh /folder/folder1` untuk menilai ukuran folder tertentu yang menurut Anda mungkin besar dan pertimbangkan untuk menghapus file lain untuk mengosongkan ruang. Salah satu opsi adalah mempertimbangkan untuk menghapus gambar Docker yang tidak digunakan dengan menjalankan `docker rmi`. Anda harus menyadari bahwa Docker memiliki ruang terbatas di lingkungan, untuk informasi lebih lanjut tentang Docker, lihat panduan Dokumentasi [Docker](#).

docker push waktunya habis dan terus mencoba lagi

Masalah: Ketika Anda `docker push` menjalankannya adalah waktu habis dan terus mencoba lagi tanpa hasil.

Penyebab: Ini dapat disebabkan sebagai akibat dari izin yang hilang, mendorong ke repositori yang salah atau kurangnya otentikasi.

Solusi: Untuk mencoba dan menyelesaikan masalah ini, pastikan Anda mendorong ke repositori yang benar. Jalankan `docker login` untuk mengautentikasi dengan benar. Pastikan Anda memiliki semua izin yang diperlukan untuk mendorong ke repositori Amazon ECR.

Peramban yang didukung untuk AWS CloudShell

Tabel berikut mencantumkan peramban yang didukung untuk AWS CloudShell.

Dukungan peramban web

Peramban	Versi
Google Chrome	Tiga versi terakhir
Mozilla Firefox	Tiga versi terakhir
Microsoft Edge	Tiga versi terakhir
Apple Safari untuk macOS	Dua versi terakhir

AWS Wilayah yang Didukung untuk AWS CloudShell

Bagian ini mencakup daftar Wilayah yang didukung dan AWS Wilayah Keikutsertaan untuk AWS CloudShell. Untuk daftar titik akhir AWS layanan dan kuota untuk CloudShell, lihat [AWS CloudShell halaman](#) di Referensi Umum Amazon Web

Berikut ini adalah AWS Wilayah yang didukung untuk AWS CloudShell:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Osaka)
- Asia Pasifik (Seoul)
- Asia Pasifik (Sydney)
- Asia Pasifik (Singapura)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Stockholm)
- Amerika Selatan (Sao Paulo)

GovCloud Daerah

Berikut ini adalah GovCloud Wilayah yang didukung untuk CloudShell:

- AWS GovCloud (AS-Timur)
- AWS GovCloud (AS-Barat)

Wilayah Keikutsertaan

Wilayah Keikutsertaan tidak diaktifkan secara default. Anda harus mengaktifkan Wilayah ini secara manual untuk menggunakannya. Untuk informasi selengkapnya, lihat [Mengelola Wilayah AWS](#).

Berikut ini adalah Wilayah Keikutsertaan yang didukung untuk CloudShell:

- Afrika (Cape Town)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Jakarta)
- Eropa (Milan)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)

Wilayah yang Didukung untuk Docker

Lingkungan AWS CloudShell komputasi hanya mendukung kontainer Docker di Wilayah berikut:

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- US West (Oregon)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Sydney)
- Asia Pasifik (Singapura)
- Asia Pasifik (Tokyo)
- Kanada (Pusat)
- Eropa (Frankfurt)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Amerika Selatan (São Paulo)

Kuota layanan dan pembatasan untuk AWS CloudShell

Halaman ini menjelaskan kuota Layanan dan pembatasan yang berlaku untuk bidang-bidang berikut:

- [Penyimpanan tetap](#)
- [Penggunaan bulanan](#)
- [Ukuran perintah](#)
- [Kerang bersamaan](#)
- [Sesi shell](#)
- [Akses jaringan dan transfer data](#)
- [File sistem dan isi ulang halaman](#)

Penyimpanan tetap

Dengan AWS CloudShell, Anda memiliki penyimpanan persisten 1 GB untuk masing-masing Wilayah AWS tanpa biaya. Penyimpanan persisten terletak di direktori home Anda (\$HOME) dan bersifat pribadi untuk Anda. Tidak seperti sumber daya lingkungan sementara yang didaur ulang setelah setiap sesi shell berakhir, data di direktori home Anda tetap ada di antara sesi.

Jika Anda berhenti menggunakan AWS CloudShell dalam Wilayah AWS, data disimpan di penyimpanan persisten Wilayah itu selama 120 hari setelah akhir sesi terakhir Anda. Setelah 120 hari, kecuali Anda mengambil tindakan, data Anda secara otomatis dihapus dari penyimpanan persisten Wilayah tersebut. Anda dapat mencegah penghapusan dengan meluncurkan AWS CloudShell lagi dalam hal itu Wilayah AWS. Untuk informasi selengkapnya, lihat [Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell](#).

Note

Skenario penggunaan

Márcia telah digunakan AWS CloudShell untuk menyimpan file di direktori rumahnya dalam dua Wilayah AWS: US East (N. Virginia) dan Eropa (Irlandia). Dia kemudian mulai menggunakan AWS CloudShell secara eksklusif di Eropa (Irlandia) dan berhenti meluncurkan sesi shell di US East (N. Virginia).

Sebelum batas waktu untuk menghapus data di AS Timur (Virginia Utara), Márcia memutuskan untuk mencegah direktori rumahnya didaur ulang dengan meluncurkan AWS

CloudShell dan memilih Wilayah AS Timur (Virginia Utara) lagi. Karena dia terus menggunakan Eropa (Irlandia) untuk sesi shell, penyimpanan persistennya di Wilayah itu tidak terpengaruh.

Penggunaan bulanan

Ada kuota penggunaan bulanan AWS CloudShell untuk masing-masing Wilayah AWS di Akun AWS. Jika Anda mencoba mengakses AWS CloudShell setelah Anda mencapai kuota bulanan untuk Wilayah tersebut, pesan akan ditampilkan untuk menjelaskan mengapa lingkungan shell tidak dapat dimulai.

Note

Jika Anda perlu meningkatkan kuota penggunaan bulanan, hubungi [AWS Support](#).

Ukuran perintah

Ukuran perintah tidak boleh melebihi 65412 karakter.

Note

Jika Anda berniat menjalankan perintah yang melebihi 65412 karakter, maka buat skrip dengan bahasa pilihan Anda, lalu jalankan dari antarmuka baris perintah. Untuk informasi lebih lanjut tentang berbagai perangkat lunak pra-instal yang dapat diakses dari antarmuka baris perintah, lihat [Perangkat lunak pra-instal](#).

Untuk melihat sebagai contoh cara membuat skrip, dan kemudian menjalankannya dari antarmuka baris perintah, lihat [Tutorial: Memulai dengan AWS CloudShell](#).

Kerang serentak

- Kerang serentak: Anda dapat menjalankan hingga 10 cangkang pada saat yang sama di masing-masing Wilayah AWS untuk akun Anda.

Sesi Shell

- Sesi tidak aktif: AWS CloudShell adalah lingkungan shell interaktif—jika Anda tidak berinteraksi dengannya menggunakan keyboard atau penunjuk selama 20-30 menit, sesi shell Anda akan berakhir. Menjalankan proses tidak dihitung sebagai interaksi.
- Sesi jangka panjang: Sesi shell yang berjalan terus menerus selama kurang lebih 12 jam secara otomatis berakhir meskipun pengguna berinteraksi secara teratur dengannya selama periode tersebut.

Akses jaringan dan transfer data

Batasan berikut berlaku untuk lalu lintas inbound dan keluar dari AWS CloudShell lingkungan Anda:

- Outbound: Anda dapat mengakses internet publik.
- Inbound: Anda tidak dapat mengakses port masuk. Tidak ada alamat IP publik yang tersedia.

Warning

Dengan akses ke internet publik, ada risiko bahwa pengguna tertentu mungkin mengekspor data dari AWS CloudShell lingkungan. Kami menyarankan agar administrator IAM mengelola daftar izin AWS CloudShell pengguna tepercaya melalui alat IAM. Untuk informasi tentang bagaimana pengguna tertentu dapat ditolak akses secara eksplisit, lihat [Mengelola tindakan yang diizinkan dalam AWS CloudShell menggunakan kebijakan khusus](#).

Transfer data: Mengunggah dan mengunduh file ke dan dari AWS CloudShell mungkin lambat untuk file besar. Atau, Anda dapat mentransfer file ke lingkungan Anda dari bucket Amazon S3 menggunakan antarmuka baris perintah shell.

Pembatasan pada file sistem dan isi ulang halaman

- File sistem: Jika Anda salah memodifikasi file yang diperlukan oleh lingkungan komputasi, Anda mungkin mengalami masalah saat mengakses atau menggunakan AWS CloudShell lingkungan. Jika ini terjadi, Anda mungkin perlu [menghapus direktori home Anda](#) untuk mendapatkan kembali akses.

- Muat ulang halaman: Untuk memuat ulangAWS CloudShell antarmuka, gunakan tombol refresh di browser Anda, bukan urutan tombol pintas default untuk sistem operasi Anda.

Riwayat dokumen untuk Panduan Pengguna AWS CloudShell

Update terbaru

Tabel berikut menjelaskan perubahan penting pada Panduan AWS CloudShell Pengguna.

Perubahan	Deskripsi	Tanggal
Tutorial baru telah ditambahkan ke Panduan AWS CloudShell Pengguna	Dua tutorial baru telah ditambahkan yang merinci cara membangun wadah Docker di dalamnya AWS CloudShell dan mendorongnya ke repositori Amazon ECR, dan cara menerapkan fungsi Lambda melalui AWS CDK	Desember 27, 2023
Kontainer Docker didukung AWS CloudShell di Wilayah tertentu	Support untuk kontainer Docker dengan AWS CloudShell telah ditambahkan di Wilayah tertentu.	Desember 27, 2023
AWS CloudShell telah bermigrasi untuk sekarang menggunakan Amazon Linux 2023 (AL2023)	AWS CloudShell sekarang menggunakan AL2023 dan telah bermigrasi dari Amazon Linux 2.	Desember 4, 2023
Wilayah AWS Baru untuk AWS CloudShell	AWS CloudShell sekarang umumnya tersedia di AWS Wilayah berikut: <ul style="list-style-type: none">• AS Barat (California Utara)• Afrika (Cape Town)• Asia Pasifik (Hong Kong)• Asia Pasifik (Osaka)	Juni 16, 2023

- Asia Pasifik (Seoul)
- Asia Pasifik (Jakarta)
- Asia Pasifik (Singapura)
- Eropa (Paris)
- Europe (Stockholm)
- Europe (Milan)
- Timur Tengah (Bahrain)
- Timur Tengah (UEA)

[Luncurkan AWS CloudShell di Console Toolbar](#)

Luncurkan CloudShell di Console Toolbar, di kiri bawah konsol dengan memilih CloudShell.

Maret 28, 2023

[AWS Wilayah Baru untuk AWS CloudShell](#)

AWS CloudShell sekarang tersedia di AWS Wilayah berikut:

Oktober 6, 2022

- Kanada (Pusat)
- Eropa (London)
- Amerika Selatan (Sao Paulo)

[AWS CloudShell didukung di AWS AS GovCloud](#)

AWS CloudShell sekarang didukung di Wilayah AWS GovCloud (AS).

Juni 29, 2022

[FAQ Keamanan](#)

FAQ tambahan berfokus pada masalah keamanan.

April 14, 2022

[Soket Web](#)

Menambahkan bagian ke persyaratan jaringan CloudShell yang menjelaskan penggunaan WebSocket protokol.

Maret 21, 2022

Memecahkan masalah tombol panah di PowerShell	Ikuti langkah-langkah untuk memperbaiki tombol panah yang salah menampilkan huruf saat ditekan.	Februari 7, 2022
Pelengkapan otomatis tombol tab	Dokumentasi baru yang menjelaskan cara menggunakan bash-completion, yang memungkinkan pelengkapan otomatis perintah atau argumen yang diketik sebagian dengan menekan tombol Tab.	September 24, 2021
Menentukan Wilayah AWS	Dokumentasi tentang menentukan default Wilayah AWS untuk AWS CLI perintah.	11 Mei 2021
Memformat dalam versi PDF dan Kindle	Ukuran gambar tetap dan teks dalam sel tabel.	10 Maret 2021
Rilis ketersediaan umum (GA) AWS CloudShell di AWS Wilayah tertentu	<p>AWS CloudShell sekarang umumnya tersedia di AWS Wilayah berikut:</p> <ul style="list-style-type: none">• AS Timur (Ohio)• AS Timur (Virginia Utara)• AS Barat (Oregon)• Asia Pasifik (Tokyo)• Eropa (Irlandia)• Asia Pasifik (Mumbai)• Asia Pasifik (Sydney)• Eropa (Frankfurt)	15 Desember 2020

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.